

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»
УДК 004.043

«До захисту допущено»

Завідувач кафедри
_____ І.Р. Пархомей
(підпис)

“ ____ ” _____ 2019 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 126 «Інформаційні системи та технології»

на тему: _____ «Система інтелектуального аналізу голосових команд, їх інтерпретація для подальшого планування дій робота»

Виконав: студент другого курсу, групи ІК-82мп
(шифр групи)

_____ Менчинський Олександр Сергійович _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник доцент, к.т.н., доцент Тимошин Ю.А. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант НК к.т.н., доцент, Пасько В.П. _____
(назва розділу) (науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ І.Р. Пархомей

(підпис)

« ____ » _____ 2019 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Менчинський Олександр Сергійович

(прізвище, ім'я, по батькові)

1. Тема дисертації «Система інтелектуального аналізу голосових команд, їх інтерпретація для подальшого планування дій робота» _____,
науковий керівник дисертації Тимошин Юрій Афанасійович, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «28»10 2019 р. № 3770-с

2. Термін подання студентом дисертації _____ 18.11.2019

3. Об'єкт дослідження: архітектура системи інтелектуального аналізу та інтерпретації голосових команд у робота.

4. Предмет дослідження: технології розробки архітектури системи інтелектуального аналізу та інтерпретації голосових команд у робота.

5. Перелік завдань, які потрібно розробити: аналіз предметної області Інтернету речей, розумного дому та розпізнавання голосу, порівняльний аналіз існуючих аналогів, технологічні принципи розпізнавання голосу, опис алгоритму роботи системи, опис структури системи, створення ПЗ системи розпізнавання голосових команд.

6. Орієнтовний перелік ілюстративного матеріалу шість плакатів

7. Орієнтовний перелік публікацій: «Дослідження можливостей технологій нейронних мереж у системах розпізнавання мовлення» // Perspective issues of world science №13 2019, «Методи розпізнавання мовлення та їх класифікація» // Perspective issues of world science №11 2019

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
НК	Пасько В.П., доцент		
Перевірка на співпадіння	Лісовиченко О.І., доцент		

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Формування проблематики	02.09.2019 – 08.09.2019	
2	Аналіз проблематики	09.09.2019 – 15.09.2019	
3	Аналіз наявних рішень	16.09.2019 – 22.09.2019	
4	Формування вимог до структури системи	23.09.2019 – 29.09.2019	
5	Опис алгоритму роботи системи	30.09.2019 – 06.10.2019	
6	Вибір технології розробки ПЗ	07.10.2019 – 13.10.2019	
7	Розробка ПЗ	14.10.2019 – 20.10.2019	
8	Тестування та покращення ПЗ	21.10.2019 – 27.10.2019	

Студент

(підпис)

Менчинський О.С.

(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

Тимошин Ю.А.

(ініціали, прізвище)

АНОТАЦІЯ

У даній роботі розглянуто проблему побудови системи інтелектуального аналізу голосових команд та їх інтерпретації для подальшого планування дій робота в системі «розумний дім».

В розділі аналізу проблематики та існуючих рішень було розглянуто важливість голосових інтерфейсів, їх застосування, проаналізовано існуючі аналоги роботів-асистентів із вбудованими підсистемами розпізнавання мовлення та окремими модулями, інструментами з даним функціоналом.

Розділ аналізу інформаційного забезпечення присвячений опису існуючих методів розпізнавання мовлення, а також опису існуючих інструментів для вирішення проблеми інтелектуального аналізу голосових команд. Також описано вимоги до функціоналу та алгоритм роботи системи.

У розділі опису розробки системи розглянуто різні інструменти для розробки, визначено спосіб представлення даних. Розподілено систему на підмодулі.

В розділі маркетингового аналізу стартап-проекту проаналізовано поточну ситуацію на ринку, розроблено стратегії та маркетингові плани для впровадження даного рішення.

Загальний об'єм роботи – 88 сторінок, 21 рисунок, 27 таблиць, 2 додатки, 21 бібліографічне найменування за переліком посилань.

Ключові слова: розпізнавання мовлення, голосові команди, робот-асистент, методи розпізнавання.

ABSTRACT

Diploma project examines the problem of building a system of intellectual analysis of voice commands and their interpretation for further planning of work in the system "smart home".

In the section of analysis of problems and existing solutions the importance of voice interfaces, their application was considered, the existing analogs of robotic assistants with integrated speech recognition subsystems and individual modules, tools with this functionality were analyzed.

The Information Analysis section is dedicated to describing existing speech recognition methods, as well as describing existing tools to solve the problem of voice command mining. Functional requirements and system algorithm are also described.

The system development section describes various development tools and defines how data is presented. The system is divided into submodules.

The Marketing Analysis section of the startup project analyzes the current market situation, develops strategies and marketing plans for implementing this solution.

Explanatory note size – 88 pages, 21 drawings, 27 tables, 2 appendices, 21 bibliographic titles by the list of references.

Keywords: speech recognition, voice commands, robot assistant, recognition methods.

Пояснювальна записка
до магістерської дисертації

на тему: *Система інтелектуального аналізу голосових команд, їх
інтерпретація для подальшого планування дій робота*

Київ – 2019 року

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	9
ВСТУП	10
1. АНАЛІЗ ПРОБЛЕМАТИКИ ТА ІСНУЮЧИХ АНАЛОГІВ.....	12
1.1. Огляд предметної області	12
1.2. Голосове управління системою «розумний дім»	13
1.3. Розпізнавання мовлення.....	15
1.4. Аналіз існуючих рішень.....	19
Висновки до розділу	28
2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ	30
2.1. Вимоги до структури та функціоналу	30
2.2. Класифікація систем розпізнавання мовлення	36
2.3. Опис алгоритмів та методів розпізнавання мовлення	37
Висновки до розділу	52
3. ОПИС РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	54
3.1. Вибір середовища та засобів розробки	54
3.1.1. Вибір мови програмування	54
3.1.2. Вибір бази даних	58
3.1.3. Вибір бібліотеки та інструментів розпізнавання мовлення.....	59
3.2. Програмні модулі	59
3.2.1. Модуль розпізнавання мовлення	60
3.2.2. Модуль користувацького інтерфейсу	62
Висновки до розділу	68

4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ	70
4.1. Опис ідеї проекту.....	72
4.2. Технологічний аудит ідеї проекту	73
4.3. Аналіз ринкових можливостей.....	74
4.4. Розроблення ринкової стратегії проекту	80
4.5. Розроблення маркетингової програми стартап-проекту	82
Висновки до розділу	84
ВИСНОВКИ.....	85
ПЕРЕЛІК ПОСИЛАНЬ	87
ДОДАТКИ.....	88

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

АОК – аналіз основних компонент

БД – база даних

НМ – нейронна мережа

ПЗ – програмне забезпечення

РД – розумний дім

ВСТУП

Актуальність проекту заключається у виявленні недоліків існуючих рішень систем інтелектуального аналізу голосових команд та проектуванні такої системи, що забезпечить більш інтуїтивну та просту взаємодію між користувачами розумного дому та роботами-компаньйонами, використовуючи інструментарій розпізнавання мовлення для перетворення звукового сигналу на інформацію у текстовому вигляді та подальшої обробки й інтерпретації повідомлення на команду виконання задачі. Ця система буде простою у користуванні, доступною, легкою в обслуговуванні та розширенні.

Метою дисертації є підвищення ефективності та спрощення взаємодії користувача з роботом-асистентом розумного дому за рахунок використання методу розпізнавання мовлення.

Об'єктом дослідження є безпосередньо архітектура самої системи розпізнавання голосових команд у робота, а предметом виступають ті технології, що використовуватимуться для її втілення.

Наукова новизна полягає у виявленні недоліків існуючих рішень та розробці системи відслідковування звукових повідомлень, що краще працює для їх обробки до легкого у маніпуляції формату і наступного перетворення на програмний сигнал про виконання певного типу відомих задач, ніж аналоги.

Мова є природною формою комунікації для людини. Інтерфейс з функцією керування голосом може поліпшити існуючий призначений для користувача інтерфейс – він забезпечує більш зручний і менш обмежений спосіб взаємодії людини з технікою. Якісний голосовий інтерфейс допомагає долати неприйняття технології користувачами, так як для його використання не потрібно опановувати нові навички.

Важливий і практичний аспект даних проблем пов'язаний з тим, що голосовий інтерфейс є необхідною складовою, коли мова йде про створення комфортних умов життя для людей з порушеннями опорно-рухового апарату, а також фахівцям, що втратив можливість використовувати стандартні засоби

в результаті професійного захворювання, травми або каліцтва . Такі системи з часом увійдуть в повсякденний побут в процесі реалізації концепції так званих «розумних будинків».

Сучасність теми дослідження визначається тим, що ринок програмного забезпечення розпізнавання голосу стрімко розвивається, охоплюючи практично всі сфери нашого життя. За даними останніх звітів на сьогоднішній день світовий ринок цих технологій оцінюється приблизно в 900 млн. доларів, а зростання становить близько 28% в рік.

У зв'язку з вищесказаним стає актуальною проблема можливості створення голосових інтерфейсів для фахівців і систем надають такі кошти, враховуючи неможливість миттєвого переходу і необхідність адаптації до нових інструментів і засобів, очевидною стає потреба в інтеграції з уже існуючими системами. На практиці рішення такого завдання і створення необхідної системи, що може бути інтегрована, виявляється нетривіальним.

Далеко не всі завдання розробки голосового інтерфейсу в даний час можна вважати вирішеними. Розробка програмного забезпечення голосового інтерфейсу є складною і комплексною. Дуже часто вона знань у великій кількості предметних областей. Серед них звичайно комп'ютерні науки, лінгвістика і навіть психологія.

1. АНАЛІЗ ПРОБЛЕМАТИКИ ТА ІСНУЮЧИХ АНАЛОГІВ

1.1.Огляд предметної області

Історія розумного будинку почалася ще в XIX столітті при створенні перших систем життєзабезпечення. Всі існуючі на сьогоднішній день технології ґрунтуються на розробках, які були створені XX століття. Ці технології повільно видозмінювалися в міру просування вперед технічного прогресу. Оснащення будинку необхідним технічним обладнанням призвело до сучасного вигляду житлового будинку: кухня з вбудованим обладнанням, включаючи газову або електричну плиту і холодильник, ванна і туалет з сучасною сантехнікою і т.д.

Подальше вдосконалення домашнього обладнання призвело до впровадження в обіг телефону, радіо і телебачення. Цей процес протікав одночасно зі створенням проектів майбутнього житла з принциповою зміною уявлення про нього. Поняття «розумного будинку» пішло від англійського терміну *intelligentbuilding* [1].

Внаслідок неправильного перекладу в німецькій мові це поняття набуло назву «розумний дім», що часто вводить в оману. Йдеться про раціональне використання агрегату «будинок», не зачіпаючи поняття «машина». При цьому основний акцент робиться на його інтерактивність.

Ця інтерактивна система складається з кількох різних мереж, таких як електромережу, телефонна і телевізійна мережі. На даний момент відбувається процес об'єднання мереж, створюється оптимальний інтерфейс між мережею та кінцевим пристроєм і, головне, інтерфейс між людиною і машиною у вигляді графічного призначеного для користувача інтерфейсу, сенсорного екрану або просто ідентифікації голосу або жестів.

Основні функції розумного будинку включають в себе управління наступними системами :

- інфраструктура життєзабезпечення (управління освітленістю, управління опаленням, управління мікрокліматом);
- системи безпеки (датчики руху, сигналізація, блокування вікон і дверей, імітація присутності, оповіщення про аварії в будинку);
- побутова техніка (управління телевізором, управління холодильником, управління чайником / кавоваркою, підтримка функцій контролю і управління через інтернет);
- енергетика (контроль альтернативних джерел енергетики);
- обслуговування (повідомлення про несправності (включаючи відправку повідомлень і листів), моніторинг і управління (включаючи взаємодію через інтернет).

1.2.Голосове управління системою «розумний дім»

Фахівці звернули увагу, що більшість користувачів системи «розумний дім» віддає перевагу такому способу регулювання її параметрів, як голосове управління. Найчастіше, причиною вибору стає практичність власників житла, адже в коло функціональних обов'язків системи можна включити будь-які елементи «Розумного будинку»:

- клімат контроль;
- безпеку та відеоспостереження;
- водопровід;
- освітлення;
- зрошення ділянки;
- басейн;
- жалюзі;
- телевізор.



Рисунок 1.1. Приклад системи «розумний дім»

Установка системи забезпечує її користувачам безліч переваг:

- використовуючи тільки запрограмовані або самостійно закладені вами голосові команди, ви тримаєте під контролем практично всі сфери свого побуту;
- застосування в створенні «розумного дому» інноваційних технологій робить його надійним і довговічним;
- система голосового управління «розумний дім» забезпечує додаткову безпеку вашого житла, так як здатна розпізнавати голоси;
- доступна вартість.

Для введення в програму нової команди немає необхідності використовувати функцію аудіозапису, досить просто відтворити її у вигляді тексту. Система сама обробить інформацію і почне виконувати відповідні команди [2]. Можна також ввести обмеження на виконання однієї або декількох команд в різних кімнатах.

Наприклад, ви хочете мати можливість керувати світлом в приміщенні, коли у вас зайняті або забруднені руки або ваша дитина ще не може дотягнутися до вимикача. В першу чергу вам потрібно точно знати потужність ламп, які будуть забезпечувати освітленість кімнат. Для впровадження голосової системи управління «Розумним домом» знадобляться:

- запобіжники;
- потужні датчики контролю над різними зонами освітлення (в приватному будинку є приміщення, які використовуються менш інтенсивно, зокрема, підвал);
- обмежувач - він актуальний при установці системи в дитячій кімнаті (так дитина не зможе балуватися зі світлом);
- мікрофони. Їх можна монтувати вимикач. Найбільш придатними є ті пристрої, які мають слабку сприйнятливостю до фонових шумів;
- контролери - для обробки надходять голосових команд.

Важливо визначити також оптимальний для всіх домочадців спосіб аудіюуправління. Буде це голосова команда або хлопки в долоні, залежить від особливостей навколишнього оточення. Наприклад, якщо у вас вдома діти, які голосно і весело спілкуються між собою, то краще вибрати для управління плескання в долоні.

1.3.Розпізнавання мовлення

В даний час все більший інтерес набувають системи розпізнавання мовлення, що обумовлено значним поширенням мобільних пристроїв. Застосування голосового управління до даних пристроїв, з огляду на обмеженість інтерфейсів ручного введення призвело б до розширення їх можливостей і спрощення у використанні.

Існуючі технології розпізнавання, як правило, не мають широкого поширення зважаючи на обмеження їх по одному або декільком параметрам. Наприклад, основна частка систем, що дозволяють розпізнати більшість

призначених для користувача запитів вимагають підключення до мережі Інтернет (до віддалених серверів, на яких відбувається обробка запитів), оскільки недостатньо їх обчислювальної потужності і обмеженості пам'яті, виділеної під словник.

Станом на 2012 рік обсяг ринку систем розпізнавання мовлення оцінюється в мільярд доларів США і планується його зростання. Великим попитом користується голосова біометрія в судово-медичних та військових цілях і є основним «драйвером» ринку [3].

Одними з перших програм для комерційного використання були Voice Navigator, Dragon NaturallySpeaking і інші. Початок використання таких програм датується початком дев'яностих років. В основному, їх використовують люди з обмеженими можливостями, набір тексту великого обсягу для яких є проблематичним. Перетворюючи голосове повідомлення диктора в текст, дані програми дозволяють не використовувати руки при друкуванні. Якість перекладу у даних програм досить низька, проте з часом вона істотно поліпшувалася.

В результаті удосконалення продуктивності мобільних пристроїв стало можливим застосування в додатках до них технологій по розпізнаванню мовлення. Прикладом такого програмного забезпечення є Microsoft Voice Command. Ця програма надає багатий функціонал можливості управління голосом. Серед наявних функцій: створення нового документа, набір повідомлення, відтворення аудіо файлів.

Можна знайти наступне застосування розпізнаванню мовлення в повсякденному житті. Прикладна демонстрація вирішення можливих проблем – доктору в поліклініці замість того щоб заповнювати вручну медичну картку хворого досить просто проговорити діагноз і дані про хворого.

Як інший приклад можна привести систему «розумний будинок», де голосові команди використовуються для управління електрикою в будинку, наприклад, включенням або виключенням світла, кондиціонером, і т.д. Так, у багатьох додатках на сучасних мобільних пристроях все більше

застосовуються системи автоматичного розпізнавання мовлення. Важливим є те, що дані застосовуються системи, незалежні від диктора і здатні розпізнавати акустичні сигнали зі спектром людського мовлення.

Наступним етапом у впровадженні технологій розпізнавання мовлення відзначається застосування інтерфейсів SSI. Дані системи засновані на процесі оброблювання мови на початковій стадії артикулювання.

Однак можна виділити значний недолік сучасних систем розпізнавання мовлення, це: занадто висока чутливість до зашумлення, тобто необхідно досить розбірливо і чітко вимовляти голосові команди вчасно зверненні до системи з розпізнавання. Треба відзначити, що SSI заснований на підході що полягає у використанні сенсорів не схильних до впливу шумів в якості доповнення до оброблених акустичних сигналів.

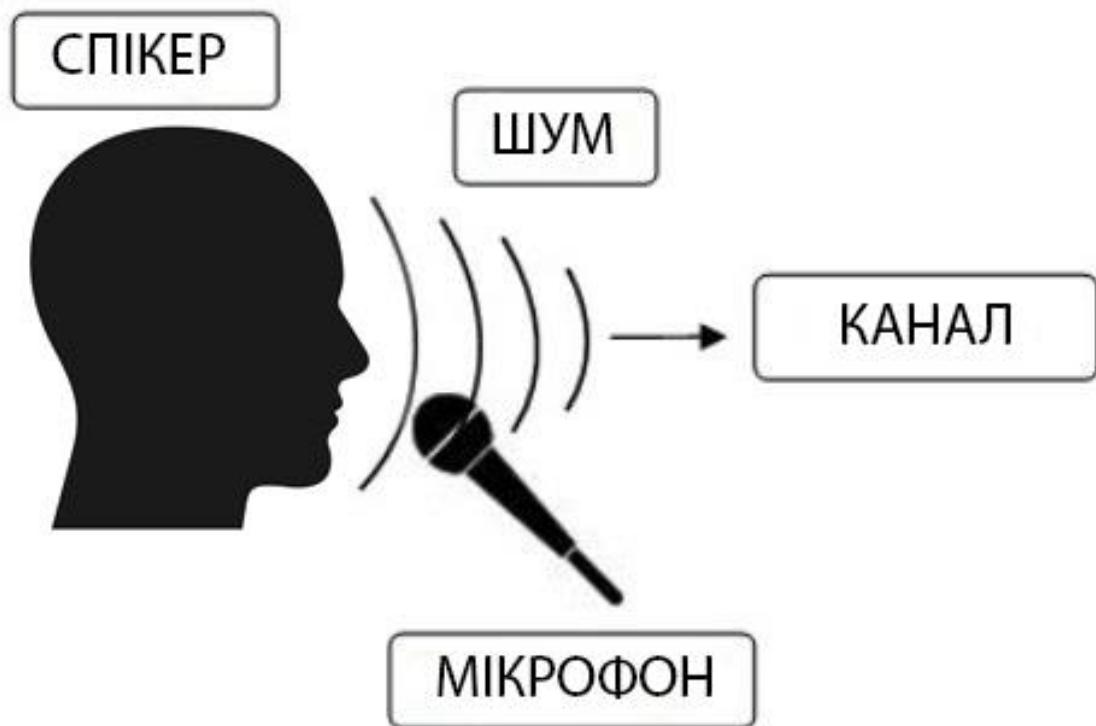


Рисунок 1.2. Зашумлення каналу під час розпізнавання мовлення

Особливості мовлення конкретної людини (через вік, стать), діалект, швидкість мовлення, фонетичний або прозодичний контекст, фоновий шум, характеристики мікрофона, направлення мови тощо.

Система повинна бути стійка до цих показників. Мобільні середовища оточені навколишнім шумом. На картинці причини акустичних варіацій у мовленні.

Таблиця 1.1. Категорії поміх при розпізнаванні голосу

Категорія поміх	Потенційні поміхи
Динамік	<ol style="list-style-type: none"> 1. Стать 2. Якість голосу 3. Діалект 4. Стрес/емоційність 5. Швидкість мовлення
Мікрофон	<ol style="list-style-type: none"> 1. Електричний шум 2. Спотворення (дісторшн) 3. Напрямок мовлення
Шум	<ol style="list-style-type: none"> 1. Інші динаміки 2. Фоновий шум 3. Реверберації
Канал зв'язку	<ol style="list-style-type: none"> 1. Спотворення 2. Шум 3. Ехо 4. Випадіння

Можна виділити п'ять основних способів розпізнавання мови:

1. Розпізнавання окремих команд – в даному способі потрібно роздільно вимовляти слово, словосполучення, а потім проводиться його розпізнавання. Якість розпізнавання даного способу обмежена.

2. Розпізнавання по граматиці – в даному способі проводиться розпізнавання вже по фразам, які відповідають певному набору правил. Граматика тут задається шляхом використання стандартних XML мов, а обмін даними між системою розпізнавання і додатком здійснюється по протоколу MRCP.

3. Пошук ключових слів в потоці зливої промови – в даному способі розпізнавання мовлення проводиться по окремих ділянках мови. Тут мова цілком може бути, як відповідна набору деяких правил, так і відповідною певним правилам. В даному методі немає необхідності переводити весь текст – в ній знаходяться лише ті ділянки, які містять задані слова чи словосполучення.

4. Розпізнавання злитого мовлення на великому словнику – в даному методі, сказана фраза, дослівно перетворюється в текст. При цьому достовірність даного методу досить висока.

5. Розпізнавання мови за допомогою нейронних систем – є досить складним методом, однак на основі нейронних мереж з'являється можливість створення учнів і систем що навчаюся самі себе, що є важливою передумовою для їх застосування в системах розпізнавання (і синтезу) мови.

1.4. Аналіз існуючих рішень.

Ідея керувати системою «розумний дім» за допомогою голосових управління не нова. І у великій кількості комерційних систем так чи інакше це вже використовується.

Наприклад Siri – зручний голосовий інтерфейс, інтегрований в iPhone. Розуміє дуже велику кількість мов, а з року в рік ця кількість лише зростає. Має широкий функціонал, але через закритий протокол, не може на пряму використовуватися як контролер для голосового управління розумними пристроями. Компанія розуміючи необхідність такого функціоналу самостійно розробила додаток HomeKit, що є можна сказати «тунелем» між Siri та розумним оточенням [4].

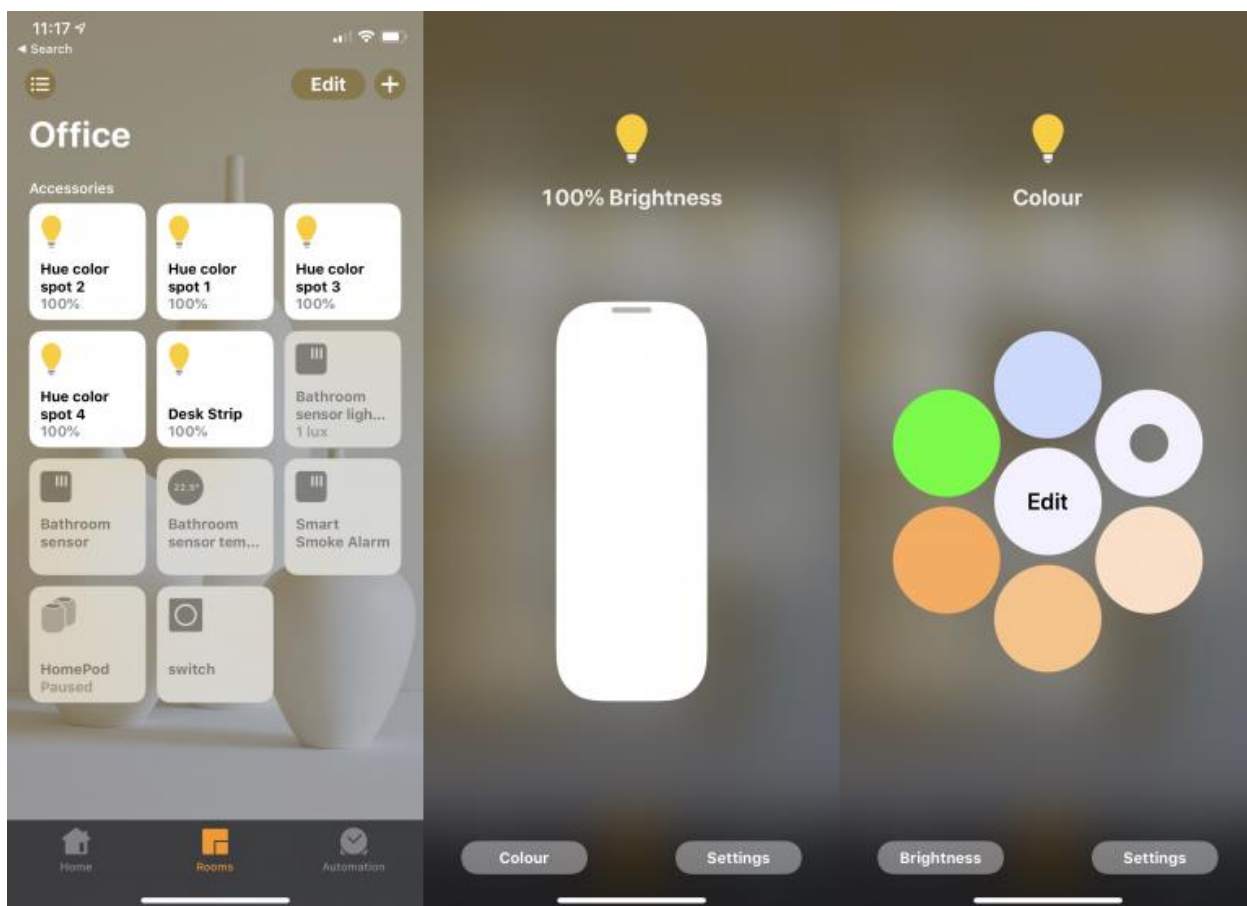


Рисунок 1.3. Інтерфейс HomeKit від Apple

До систем зі схожим функціоналом та задачами що вони вирішують можна віднести наступні:

- LG CLOi, лінійка роботів помічників з різною функціональністю. Мають голосове управління, здатен керувати іншою розумною технікою в домі.
- Asus Zenbo, робот-помічник, має здатність самостійно пересуватися по приміщенню, розпізнавати обличчя, шукати інформацію в мережі, та що найголовніше для нас – має голосове управління.
- Omote TrueSmart Robot, робот-помічник, в основі якого закладено голосового асистента Amazon Alexa, що дає змогу знаходити інформацію в мережі, спрощує керування системою «розумний дім». Оснащений 5ма мікрофонами.

Таблиця 1.2. Аналіз аналогів роботів-помічників з вбудованими системами розпізнавання мовлення

Назва	Переваги	Недоліки
Asus Zenbo	1. Самостійно пересувається по приміщенню. 2. Розпізнавання обличчя. 3. Має здатність пошуку інформації у мережі. 4. Має голосове управління.	1. Відсутність маніпуляторів. 2. Має обмежену можливість у пересуванні.
LG CLOi	1. Має голосове управління. 2. Керування іншими розумними пристроями. 3. Має функцію нотаток	1. Не пересувається самостійно.
Omote TrueSmart Robot	1. Має функцію пошуку інформації за допомогою «Алекси». 2. Керування системою «розумний дім» 3. Оснащений 5ма мікрофонами.	1. Пропрієтарне програмне забезпечення «Алекса» від Amazon
Cortana	Схожий функціонал. Різниця лише в алгоритмах, методиках навчання нейронних мереж, тощо.	
Siri		
Google Asistant		
Amazon Alexxa		

Основним недоліком розглянутих рішень є пропріитарність та те що вони прив'язані до конкретних технічних реалізацій. Перевагою нашої системи буде відкритість та здатність до розширення та подальшого вдосконалення.



Рисунок 1.4. Asus Zenbo

Звернімо увагу на готові рішення інструментів розпізнавання мовлення, що можуть бути інтегровані до більш складних систем.

Модель SpeechRecognition побудована на движку DeepSpeech з використанням однойменної архітектури Baidu і платформи машинного навчання TensorFlow від Google.

Крім моделі з розпізнавання мовлення, до набору входять модулі для Python і NodeJS, системи, що спрощують інтеграцію в програмні продукти, а також інструменти для командного рядка. Відмітними особливостями DeepSpeech є низька вимогливість до системних ресурсів (можливий запуск навіть на Raspberry Pi) і висока якість розпізнавання при сторонніх шумах.

Як заявляє Mozilla, ідея проекту полягає в тому, щоб паралельно з новою моделлю розпізнавання мовлення створити мовленнєвий корпус, з відкритим початковим кодом, достатнього об'єму для використання у великих проектах.

Таблиця 1.3. Аналіз аналогів інструментів розпізнавання мовлення

Назва	Характеристика
BitVoicer	Програма розпізнавання мовлення. Дає можливість простим пристроям, що мають низьку потужність обробки, стати голосовими. Для цього BitVoicer використовує потужність обробки ПК для аналізу звукових потоків, ідентифікації пропозицій, присутніх у цих потоках, та відправки команд до підключеного до нього мікроконтролера.
Cloud Speech-to-Text	Відкритий API для обробки аудіофайлів з метою розпізнавання голосу від компанії Google. Має обмежену кількість безкоштовних запитів. Повертає json об'єкт з інформацією про точність розпізнавання та сам текст відповідно.
SpeechRecognition	Web-інтерфейс розпізнавання мовлення від Mozilla. Перевагами є те що система є безкоштовною, може бути інтегрована до розширень браузерів.
SpeechKit	Відкритий API від компанії Yandex. Серед переваг перед Cloud Speech-to-Text те, що має більш широкий функціонал та кількість безкоштовних запитів не обмежена.
CMU Sphinx	Сфінкс CMU (скорочено як Sphinx) – загальний термін для опису групи систем розпізнавання голосу, розроблених в університеті Карнегі Меллона. Вона включає в себе серію програм розпізнавання голосу (Sphinx 2-4) та тренер з акустичної моделі (SphinxTrain).

Продовження таблиці 1.3

Speech Recognition	Інструмент для розпізнавання голосу від Microsoft. Використовується у Windows.
Snowboy Hotword Detection	Механізм виявлення "гарячих слів", який можна запускати локально. В основному він прослуховує ключову фразу (наприклад, "Привіт Олександр"), а потім встановлює прапори, що ця фраза була виявлена. Ця функція корисна. Оскільки знімає необхідність у пристрої постійно надсилати все, що він чує, на сервери Amazon LEX, щоб побачити, чи повинен він відповідати. Натомість він прослухає свою ключову фразу, а потім припустить, що все, що було сказано після цього, призначене для неї, поки не наступить період очікування або не буде виконано поточний намір.
Kaldi	Інструментарій розпізнавання мови, написаний на C++ та ліцензований під ліцензією Apache v2.0. Калді призначений для використання дослідниками розпізнавання мовлення.
Julius	Движок розпізнавання мовлення, зокрема високопродуктивний, двохпрохідний великий словник безперервного розпізнавання мовлення (LVCSR), програмне забезпечення для декодера, пов'язане з мовленням.
Wit Speech	Wit Speech дозволяє розробникам легко створювати додатки та пристрої, з якими можна говорити чи надсилати тексти. Відкрита та розширювана платформа природної мови.

Закінчення таблиці 1.3

Wit.ai	Wit.ai вивчає людську мову з кожної взаємодії та використовує спільноту: те, що вивчено, ділиться між розробниками.
VaxForge	VaxForge – це безкоштовний мовленнєвий інструмент та акустична модель сховища для движків розпізнавання мовлення з відкритим кодом. Був створений для збору транскрибованого мовлення для створення вільного інструменту мови GPL для використання з движками розпізнавання мови з відкритим кодом. Аудіо файли мовлення будуть компільовані в акустичні моделі для використання з движками розпізнавання мовлення з відкритим кодом, такими як Julius, ISIP, а також Sphinx і HTK.

Усі розглянуті інструментарії та програми мають різну функціональність, використовують різні алгоритми та методи обробки інформації та розпізнавання мовлення. В більшості всі вони базуються на дуже складних концепціях, що вимагають від розробників глибоких знань предметної області.

Майже усі рішення спрямовані на роботу з великими словниками, що є не дуже принциповим аспектом для нашої системи, адже функціонал робота-помічника в системі «розумний дім» сильно обмежений та кількість типових задач не є дуже великою.

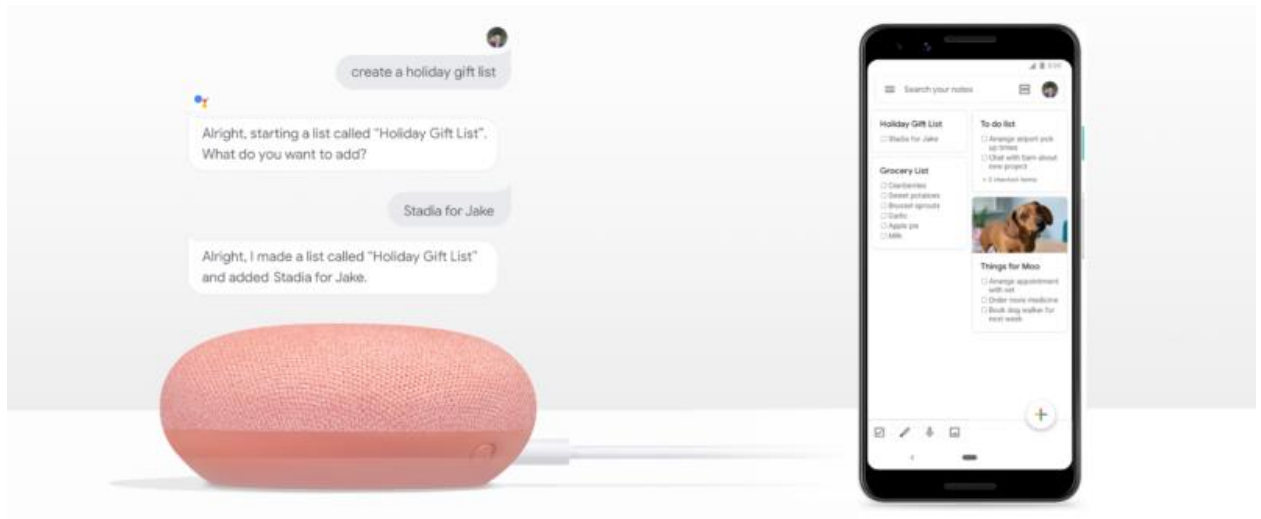


Рисунок 1.5. Додаток Google assistant

Text to speech – це штучне виробництво мови. Комп'ютеризована система, яка використовується для цієї мети, називається мовленнєвим комп'ютером або синтезатором голосу і може бути реалізована в програмних або апаратних продуктах.

Таблиця 1.4. Інструменти Text to Speech

Назва	Характеристика
Speech Synthesis	Система мовленнєвого синтезу, вільний програмний багатомовний робочий стіл із синтезу мови, який працює на декількох платформах, де пропонується відкрита архітектура для дослідження синтезу мовлення.
SpeechKyt	Інструментарій для відтворення тексту у голосовій формі від компанії Yandex.
Gusspeech	Інструментарій дозволяє легко виробляти високоякісний комп'ютерний мовний вихід, розробляти нові мовні бази даних та створювати керовані мовленнєві стимули для психофізичних експериментів.

Закінчення таблиці 1.4

eSpeenkNG	Компактний синтезатор мови з відкритим кодом для Linux, Windows та інших платформ. Він використовує метод синтезу формантів, який надає багато мов невеликим розміром. Значна частина програмного забезпечення підтримки мови здійснюється за допомогою власних файлів правил зворотного зв'язку.
IBM TTS	За допомогою тексту Watson Text-to Speech ви можете генерувати людський звук із написаного тексту. Поліпшення досвіду та взаємодії з клієнтами шляхом взаємодії з користувачами на різних мовах та тонах. Підвищити доступність контенту для користувачів з різними можливостями, надайте варіанти звуку, щоб уникнути відволікання їзди, або автоматизуйте взаємодію з обслуговування клієнтів для підвищення ефективності.
MicrosoftSpeech	Інструмент для розробників додатків від Microsoft.
Apple TTS	Інструмент для розробників додатків під операційну систему iOS.
gTTS	Бібліотека від Google для перетворення тексту в голосові форми
Pytsx	Бібліотека для мови програмування Python для textToSpeech.

Система текстового мовлення перетворює звичайну мову тексту в аудіо формат; інші системи відтворюють мовне символічне представлення у вигляді фонетичних транскрипцій у мовленні.

Синтезована мова може бути створена за допомогою об'єднання записаних фрагментів мовлення, які зберігаються в базі даних. Системи відрізняються величиною мовних одиниць, що зберігаються.

Система, яка зберігає фоно та диффоно, дозволяє ширший спектр звуків, але не має чіткості. Для конкретних цілей використання розмір повних слів або пропозицій забезпечує кращу якість звуку. Крім того, синтезатор може включати модель голосового тракту або інші характеристики людського голосу, щоб повністю відтворити "синтетичний" голос .

Висновки до розділу

Сьогодні ринок «розумних будинків» дуже стрімко розвивається. Голосове управління також вже не є чимось чим можна здивувати. В кожному сучасному смартфоні є свої рішення, що забезпечують розпізнавання та аналіз мовлення, відтворення тексту. Всі компанії також розуміють про найактуальніші сфери застосування цієї технології. Apple вже зараз має HomeKit, софт, що служить провідником між розумними девайсами та Siri.

Як ми побачили вже існує дуже велика кількість готових рішень як роботів-помічників з інтегрованою системою розпізнавання голосу так відповідно і самих підсистем, модулів або просто наборів інструментів які можна застосовувати.

Головними недоліками звичайно є те, що всі рішення є платними, а ті що безкоштовні мають обмежений функціонал або закриту систему. Проаналізувавши аналоги стало зрозуміло, що запропонована нами система має право на життя та може нести новизну.

Також нам знадобиться модуль для зворотного перетворення тексту в аудіо файл. Система текстового мовлення перетворює звичайну мову тексту в аудіо формат. Інші системи відтворюють мовленеве символічне

представлення у вигляді фонетичних транскрипцій у мовленні. Система, яка зберігає фоно, дозволяє ширший спектр звуків, але не має достатньої чіткості. Для конкретних цілей використання розмір повних слів або пропозицій забезпечує кращу якість звуку.

Інтерфейс повинен взаємодіяти з користувачем, інакше він стає не інтуїтивним. Ми також розглянули існуючі аналоги та проаналізували їх переваги та недоліки.

2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Вимоги до структури та функціоналу

Алгоритм роботи системи інтелектуального аналізу голосових команд та їх інтерпретація для подальшого планування дій робота-помічника повинен описувати повний цикл взаємодії з користувачем.

Починаючи від того в якому режимі знаходиться робот в стані спокою, в нашому випадку це буде режим очікування кодового слова як сигнал про початок обробки голосу та порівняння отриманого з базою даних типових команд, закінчуючи тим що повинен бути прописаний алгоритм поведінки в ситуаціях, коли система не змогла розібрати слово, або не знайшла відповідної команди.

Алгоритм має бути циклічним, адже після виконання роботом поставленої задачі він повинен повертатися до режиму очікування. Алгоритм також має бути стійким до помилок, адже існує багато сценаріїв в такій складній та великій системі як «розумний дім», коли неправильна інтерпретація голосових команд може призвести до небажаних наслідків або до тупикових та невизначених ситуацій. Комунікація повинна бути також інтуїтивно зрозумілою для користувача.

Потрібно описати сценарії поведінки робота в усіх можливих ситуаціях. Робот-помічник може бути вже у процесі виконання задачі. В такому випадку він повинен сповістити користувача про це та повернутися до виконання задачі.

Якщо після інтерпретації та порівняння голосової команди з існуючими, не було виявлено співпадінь – робот повинен сповістити про це користувача та не виходити з режиму прослуховування команди до очікування кодового слова чекати уточнення команди. У разі якщо користувач за будь-яких причин не повторював команду – повернутися у режим очікування.

Також буде використана система Text to Speesh, бо робот повинен багато комунікувати з користувачем та сповіщати його про свій поточний стан, про виконання команди або про некоректну інтерпретацію.

Основними компонентами системи інтелектуального аналізу голосових команд та їх інтерпретації будуть:

1. Сам інструмент прослуховування та обробки голосових команд.
2. База даних команд.
3. Веб-інтерфейс.

Потрібно обрати метод розпізнавання мовлення, мову технічної реалізації інтерфейсу та базу даних словника робота.

1. Система очікує ключове слово.
2. Система розпізнала ключове слово.
3. Система сповістила користувача звуковим сигналом про зміну стану на очікування команди.
4. Система очікує на команду.
5. Система розпізнала слово або словосполучення.
6. Система обробила та отримала текст команди.
7. Команда знайдена.
8. Робот переходить до виконання задачі відповідної до команди.
9. Система сповістила користувача про початок виконання задачі та відповідно зміну поточного стану.
10. Робот виконав команду.
11. Система сповістила про виконання задачі та відповідне повернення до стану очікування ключового слова.

Сценарій помилкового отримання ключового слова. Ключове слово повинне бути унікальним, але якщо трапилось що під час звичайної розмови було ідентифіковане системою дане слово система має сповістити про це користувача.

В такому випадку повинно бути передбачено автоматичний перехід до стану очікування, якщо після розпізнання ключового слова не було виявлено подальших акустичних сигналів.

1. Система розпізнала ключове слово.
2. Система сповістила звуковим сигналом користувача про перехід у стан очікування команди.
3. Після 10 секунд система автоматично повертається у стан очікування у разі якщо не було отримано подальших інструкцій.
4. Система проінформувала користувача про перехід до стану спокою.

Сценарій коли відповідної команди не було знайдено у базі даних. Можливий варіант, що система не змогла знайти команд з відповідною назвою. В такому випадку система повинна проінформувати користувача про це.

Головне щоб система після цього поверталася до режиму прослуховування команди, а не ключового слова. В другому випадку інтерфейс стає не інтуїтивним та занадто складним у використанні.

1. Система в стані очікування команди.
2. Система отримала команду.
3. Система не знайшла відповідної команди у базі даних.
4. Сповістила про це користувача.
5. Повернулася до стану прослуховування команди.
6. Сповістила про зміну стану користувача.

Очевидно, що при заповненні бази даних потрібно уникати команд зі схожими назвами. Як ми вже зрозуміли система розпізнавання мовлення є дуже комплексною та складною, будь які шуми або проблеми з дикцією користувача можуть причинити небажані наслідки, не потрібно ускладнювати ще більш цей процес.

Якщо система не може розпізнати слово це одна справа, але якщо система може розпізнати їх некоректно та виконувати дії, яких не хотів виконувати користувач це зовсім інше. Друге набагато гірше.

Дану проблему можна вирішувати декількома способами. Якщо система отримала результат з недостатньою точністю – вона може сповістити про це користувача та перепитати що саме він мав на увазі. Можливий варіант навіть запропонувати ці схожі варіанти.

Але це вносить складність та може дратувати користувача. Набагато більш дієвим варіантом є просте уникнення такого у базі даних.

Після того, як система розпізнала звертання до себе, вона дає користувачу голосове повідомлення про те, що слухає його команду. Далі система почне слухати господаря на предмет команди. Якщо господар надає команду, яка після розпізнавання не співпадає з жодною із відомих роботу команд, будь то через помилку господаря чи велику зашумленість, робот сповіщає, що не зміг зрозуміти, що від нього вимагає господар.

У разі, якщо команду було розпізнано, робот сповіщає господаря про успішну комунікацію та переходить до виконання дії, після чого повертається до стану прослуховування. Набір команд буде визначеним, але він може бути легко розширений, якщо для того з'явиться необхідність або попит. Клієнт (робот) та сервер (система) спілкуватимуться через API.

Нижче наведемо спрощену схему такої структурної побудови.

З функціональної ж точки зору, користувач хотів би бачити конкретний набір можливостей взаємодії та доступних команд. Із можливостей взаємодії в рамках самої системи присутні, звісно, взаємодія голосом, а також взаємодія через веб-інтерфейс.

Останній має більше опцій і налаштувань, а також дозволяє моніторити статистику та стан робота. З доступних команд можна навести такий перелік найбільш використовуваних та популярних:

1. Знайти інформацію про X у мережі.
2. Дізнатися погоду у місті X.
3. Повідомити про свій стан, положення, поточну дію.
4. Почати виконувати дію щодо розумного дому (почати прибирання, готування їжі тощо).

5. Взаємодія з іншими розумними пристроями.

Розглянемо структуру кожного окремого компонента.

Компонент прослуховування та інтерпретації голосу уявлятиме собою модульний додаток, у якого буде наявний інтерфейс взаємодії типу REST, із точками доступу у вигляді URL-посилань, що повертатимуть у відповідь на запити, з параметрами чи без, повідомлення у вигляді обробленої інформації чи необхідних даних.

Позаду знаходитиметься модуль, у якому буде реалізована головна логіка розпізнавання голосу. Створюється екземпляр класу розпізнавача голосу, який міститиме екземпляри класів аудіодрайвера, розпізнавача голосу, мікрофона та відтворювача голосу, а також стану прослуховування, поточну мову і останній розпізнаний результат.

Основною перевагами такої системи стануть її модульність та монолітність: веб-інтерфейс реалізовано окремим модулем, який отримує інформацію від окремого незалежного циклічного процесу, що запускається паралельно. Незалежність модулів покращить підтримку системи.

Він володіє методом, що виконує ініціалізацію вищевказаних параметрів, а також методом, що запускає циклічний процес розпізнавання, і повертає результат розпізнавання.

Цей модуль запускається окремим процесом, і записує результати на кожній ітерації, а також відтворює голосове повідомлення користувачу залежно від результату.

Модуль веб-інтерфейсу відповідає за взаємодію з системою на рівні веб-браузера, отримує інформацію, яку записує розпізнавач голосу, відображає її процес інтерпретації та результат, інформацію про стан розпізнавача тощо у реальному часі. Працює також окремим процесом, має адресу для доступу та є незалежним від процесу розпізнавача. Також буде реалізовано можливість подання команд не тільки в голосовій формі, а й за допомогою веб-інтерфейсу. Це може вирішити потенційні невизначені ситуації та підвищити зручність системи.

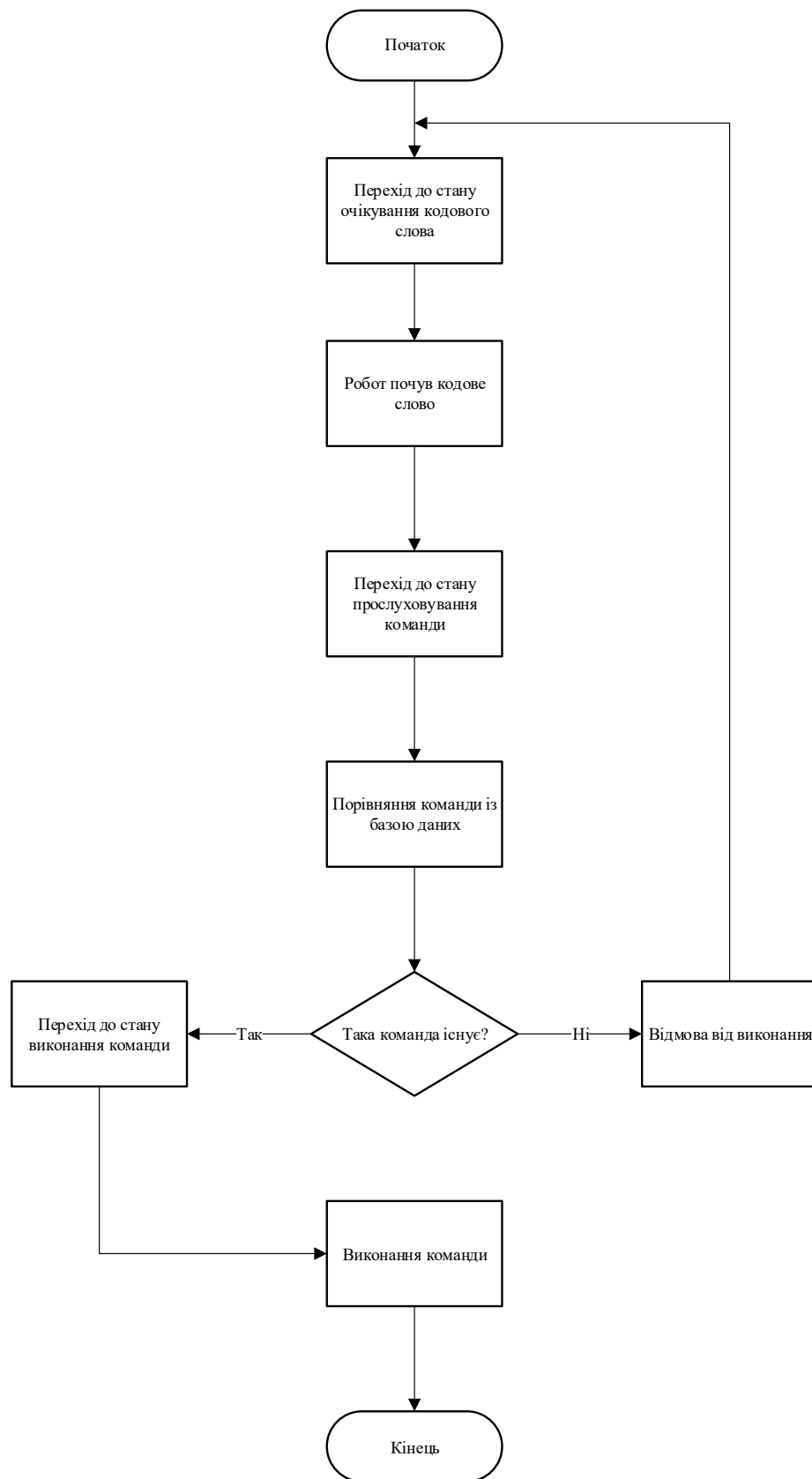


Рисунок 2.1. Алгоритм роботи системи розпізнавання

2.2.Класифікація систем розпізнавання мовлення

При розгляді класифікації систем розпізнавання мови слід відзначити, що класифікація може здійснюватися за різними параметрам. За літературними джерелами системи розпізнавання мови можна класифікувати в такий спосіб:

- в залежності від розміру словника: системи розпізнавання мови з обмеженим набором слів; системи зі словником великого розміру;
- в залежності від прив'язки до диктора: системи, що є залежними та не залежними від дикторів;
- в залежності від типу розпізнається мовлення: системи, що працюють з зливою або роздільною промовою;
- в залежності від призначення системи прийнято виділяти системи диктування і командні системи;
- в залежності від алгоритму, який використовується в системі розпізнавання виділяють: нейронні мережі, приховані Марківські моделі, динамічне програмування;
- в залежності від типу структурних одиниць, що використовуються в системі (можуть бути використані слова, фонemi, алофон, фрази);
- в залежності від принципу, за яким виділяються структурні одиниці системи розпізнавання можна розділити на системи в яких здійснюється розпізнавання за шаблоном і системи виділення лексичних елементів;

При розгляді систем розпізнавання мови слід зазначити, що в існуючих система використовується, як правило два підходи, які є принципово різними:

- аналіз голосових міток;
- аналіз елементів лексичного значення.

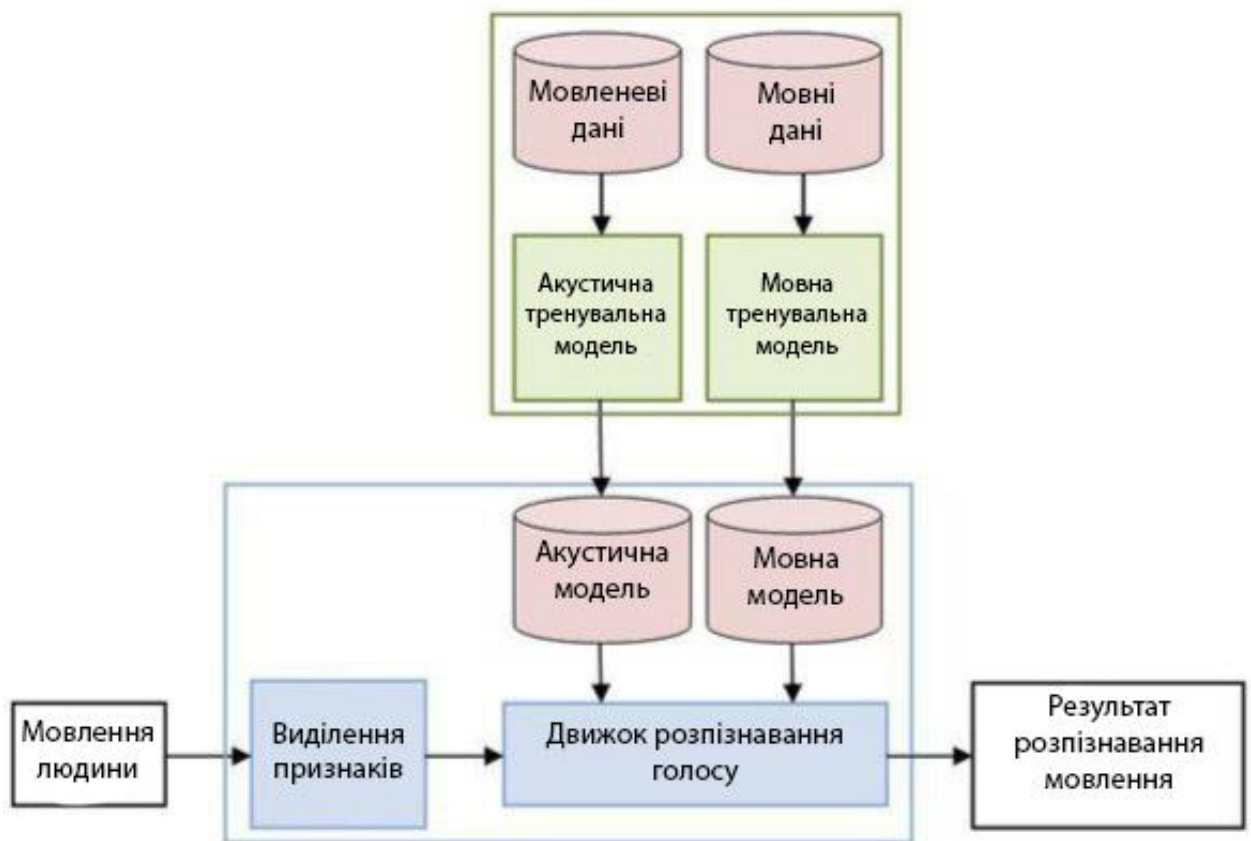


Рисунок 2.2. Архітектура системи розпізнавання мовлення

Розпізнавання голосових міток полягає в тому, що розпізнавання фрагментів мови здійснюється за зразком, який записаний заздалегідь. Даний підхід може бути використаний тільки в простих системах розпізнавання мови, які використовуються для виконання спочатку записаних голосових команд [5].

Розпізнавання лексичних елементів є більш складним підходом. Дані методи передбачають виділення з потоку мови окремих лексичних елементів, тобто фонем і алофонів. Надалі виділені лексичні елементи об'єднуються в склади і морфеми. Таким чином, в складних системах розпізнавання мови використовується саме цей підхід.

2.3.Опис алгоритмів та методів розпізнавання мовлення

До класу систем, незалежних від диктора відносяться системи, які працюють незалежно від того, хто виступає в якості диктора. Дані системи

мають можливість розпізнавання мови будь-якого диктора і не потребують попереднього навчання.

До класу систем, що залежать від диктора відносяться системи, які вимагають попереднього навчання і в його процесі налаштовуються на певного диктора [6]. При зміні диктора в таких системах виникає необхідність повної переналаштування.

Таким чином, для створення системи розпізнавання мови будь-якого класу, використовувана в промисловому масштабі необхідний багаторічний досвід в практичному застосуванні різних мовленевих технологій.

Щодо нашого конкретного випадку, нам важливо мати незалежну від диктору систему, адже вона повинна бути легко інтегрована як підсистема, та розпізнавати голос різних користувачів.

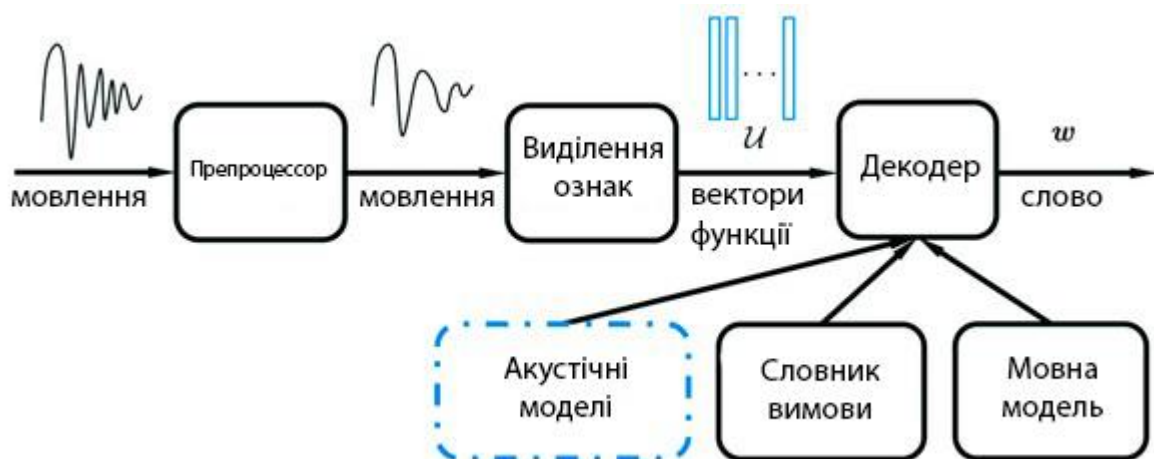


Рисунок 2.3 Схема автоматичного розпізнавання мови

Виділяються такі методи розпізнавання мовлення:

1) Розпізнавання за зразком.

Даний метод використовується в стандартному мобільному телефоні. В даному випадку система розпізнавання мови використовується з метою прискореного набору номера абонента, що викликається і списку контактів мобільного телефону.

Дана система працює таким чином. Якщо необхідно ввести новий контакт в записну книжку, то система передбачає додавання голосових позначок, яка визначає цей контакт. Як правило це прізвище і ім'я абонента,

які необхідно вимовити, можливо, для правильного внесення дану операцію доводиться робити двічі або тричі.

Далі, при необхідності виклику даного абонента, активується команда голосового набору шляхом натискання певної клавіші і вимовляється голосовою позначкою. За голосової мітки здійснюється вибір абонента і встановлюється з ним зв'язок [7].

Слід зазначити, що таке голосове управління реалізовано не тільки в мобільних телефонах, але і в інших пристроях, таки, наприклад, як комп'ютерна клавіатура.

При організації голосового управління клавіатурою, в останню вбудовується мікрофон і призначаються голосові мітки для різних клавіш, або комбінацій клавіш. Використання таких клавіатур є зручним для людей з обмеженими можливостями, а, так само, за допомогою такої клавіатури є можливість прискорити введення інформації.

Однак, при наявності шумовий складової в приміщенні, якість розпізнавання голосових міток значно погіршується.

2) Препроцесор акустичних сигналів.

Шуми і сторонні сигнали можуть стати великою проблемою. Для цього потрібно робити попередній аналіз та обробку акустичного сигналу. Під час цього процесу буде видалено частотний спектр поза спектром людського мовлення.

Цей процес виконується за допомогою смугової аналогової та цифрової фільтрації. Потім сигнал переводиться до цифрового формату [8].

Через те, що всі ці обробки можуть виконуватися центральним процесором, системи розпізнавання мовлення не потребують спеціалізованого обладнання або апаратного забезпечення.

Нормалізація вхідного акустичного сигналу є важливим кроком препроцесора. Таким чином може бути досягнуте збільшення точності розпізнавання мовлення, уникнення проблем з гучністю.

Нормалізація вхідного акустичного сигналу хоч і вирішує проблеми, але за деяких обставин можуть з'явитися проблеми у вигляді шумів. Недостатньо повний спектр або недостатня гучність призведуть до того, що інформація в сигналі може бути неповною.

3) Виділення ознак акустичного сигналу.

Завдання розпізнавання мовлення може бути ускладнене. Для вирішення поставленої задачі потрібно виділити первинні ознаки мовлення людини.

Для цього методу застосовується мовлення у спектральному вигляді. Першим кроком є смугова фільтрація та отримання частотного спектра акустичного сигналу. Виконується дискретне перетворення Фур'є.

На другому кроці отриманий частотний спектр перетворюється:

- логарифмічні масштабування частот і амплітуд;
- згладжування спектра акустичного сигналу;
- зворотне перетворення Фур'є.

Крім спектральних характеристик, необхідно враховувати і динамічні особливості мовлення. Для цього використовують похідні за часом від головних параметрів. При цьому ми можемо відстежувати не тільки зміна параметрів мови, але і швидкість їх зміни [9].

4) Виділення фонем і алофонів:

Для виділення фонем і алофонів застосовуються нейронні мережі і метод формування нейронних ансамблів. При цьому навчання виділенню примітивів мови (фонем і алофонів) може полягати у формуванні нейронних ансамблів, ядра яких відповідають найбільш частою формою кожного примітиву.

Формування нейронних ансамблів є процес навчання нейронної мережі без вчителя, при якому відбувається статистична обробка всіх сигналів, що надходять на вхід нейронної мережі. При цьому формуються ансамблі, відповідні найбільш часто зустрічається сигналам.

Запам'ятовування рідкісних сигналів відбувається пізніше і вимагає підключення механізму уваги чи іншого контролю вищого рівня.

5) Рівні розпізнавання злитого мовлення.

Розпізнавання зливої промови є багаторівневий процес. Після попередньої обробки акустичного сигналу і виділення з нього інформативних ознак виконується виділення лексичних елементів мови. Це перший рівень розпізнавання.

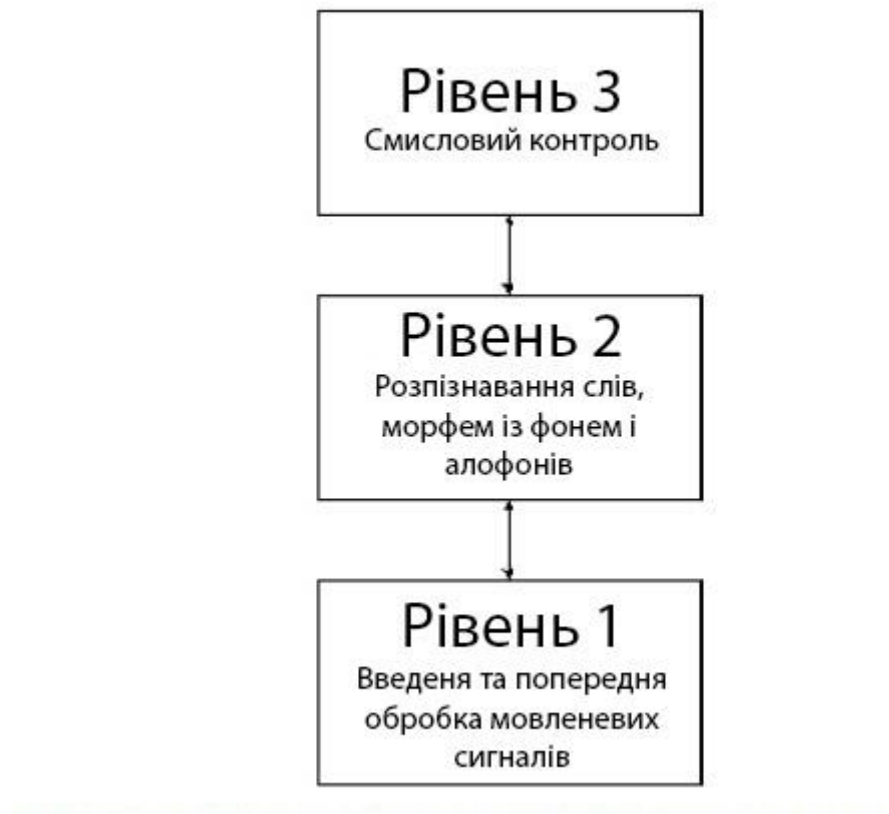


Рисунок 2.4. Рівні розпізнавання злитого мовлення

На другому рівні виділяються склади і морфеми, на третьому – слова, пропозиції та повідомлення. На кожному рівні сигнал кодується представниками попередніх рівнів. Тобто склади і морфеми складаються з фонем і алофонів, слова – з складів і морфем, пропозиції та повідомлення – зі слів [10].

При переході з рівня на рівень крім представників сигналів передаються і деякі додаткові ознаки, тимчасові залежності і відносини між сигналами. Збираючи сигнали з попередніх рівнів, вищі рівні розташовують великим об'ємом інформації (або її іншим поданням), та можуть здійснювати

управління процесами на нижчих рівнях, наприклад, з залученням механізму уваги.

Механізм уваги використовується при навчанні нейронної мережі. У випадку використання такого механізму при появі зразка, невідомого нейронній мережі, швидкість навчання багаторазово зростає.

б) Застосування нейронних мереж для розпізнавання мовлення.

При навчанні мережі з учителем можна навчити мережу розпізнавати об'єкти, належать заздалегідь певного набору класів. Якщо ж мережа навчається без учителя, то вона може групувати об'єкти за класами в відповідності до їх цифрових параметрів [11].

Таким чином, на базі нейронних мереж можна створювати системи що самі навчаються.

Також висуваються деякі вимоги до самонавчальних систем:

- розробка складається з побудови архітектури самої системи;
- можливість контролювати свої дії та в подальшому коригувати їх.

Розробник проектує тільки функціональну частину. Він не займається наповненням системи інформацією, або лише в мінімальних обсягах. Згодом у процесі навчання система отримує усю інформацію самостійно.

У цьому методі описується необхідність зворотної взаємодії в системі. Ланцюги типу «дія – результат – корекція» зустрічаються в складних біологічних організмах .

Процес навчання штучний нейронних мереж розглядається як налаштування архітектури і ваг зв'язків між нейронами (параметрів) для ефективного виконання поставлених перед мережею завдань. Існує два великих класи навчання: клас детерменірованих методів і клас стохастичних методів.

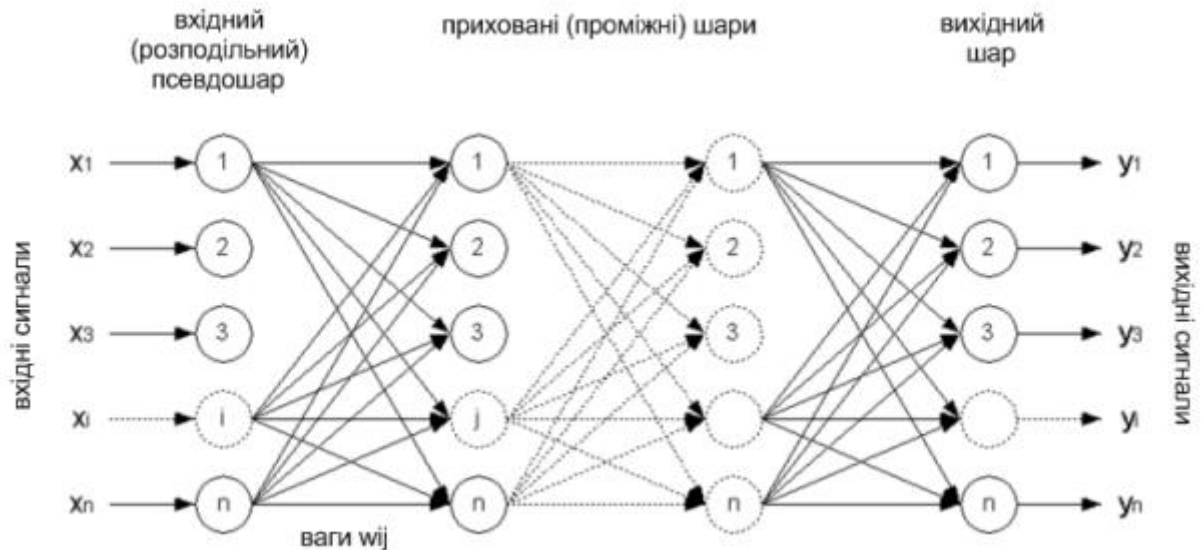


Рисунок 2.5. Навчання нейронної мережі

7) Нейронні ансамблі.

Відзначається, що в якості моделі нейронної мережі, придатної для розпізнавання мовлення і навченою без вчителя можна вибрати самоорганізуючу карту ознак Кохонена.

Нейронні ансамблі створюються для вхідних акустичних сигналів. Ще однією проблемою є мінливість мовлення, що вирішується завдяки статистичному усередненню.

У порівнянні з класичним програмуванням, коли алгоритм вирішення тієї чи іншої задачі поставлено жорстко, нейронні мережі дозволяють динамічно змінювати алгоритм простою зміною архітектури мережі.

8) Генетичні алгоритми.

Можливість зміни алгоритму роботи нейронної мережі простою зміною її архітектури дозволяють вирішувати завдання абсолютно новим способом, за допомогою так званих генетичних алгоритмів [12].

При використанні генетичних алгоритмів створюються правила відбору, що дозволяють визначити, краще або гірше справляється нова нейронна мережа з рішенням завдання. Крім того, визначаються правила модифікації нейронної мережі.

Змінюючи досить довго архітектуру нейронної мережі і відбираючи ті архітектури, які дозволяють вирішити задачу найкращим чином, рано чи пізно можна отримати вірне рішення задачі.

Генетичні алгоритми зобов'язані своєю появою еволюційній теорії. Таким чином, існує можливість створення таких нейронних мереж, які раніше не вивчалися [13].

9) Система SAS та реалізація вводу виводу.

Ця система, виконана з використанням технології нейронних мереж, призначена не тільки для розпізнавання, але і для синтезу мови [14].

При розпізнаванні мовлення система SAS здійснює введення звукової інформації, попередню обробку, отримання енергетичного спектра і виділення примітивів мови.

При синтезі мови здійснюється виділення з нейронної мережі запам'ятованого примітиву, синтез спектра (частотний параметричний синтез) і перетворення спектра в звуковий сигнал.

При навчанні послідовним повторенням двох вищеописаних процедур здійснюється запам'ятовування примітивів мови в нейронної мережі. Прийом звукового сигналу відбувається в реальному часі за допомогою звукової карти або у вигляді аудіо файлів в потрібному форматі та з визначеним кодуванням.

Попередня обробка звуку: звукові сигнали, отримані в реальному часі або введені з файлів формату WAV, піддаються в системі SAS попередній обробці [15].

10) Виконання дискретного перетворення Фур'є.

Результат згладжування Хеммінга піддається в системі SAS дискретному перетворенню Фур'є за алгоритмом швидкого перетворення Фур'є. В результаті цього отримуємо амплітудний спектр і інформацію про фазу сигналу (в реальних і уявних коефіцієнтах) [16].

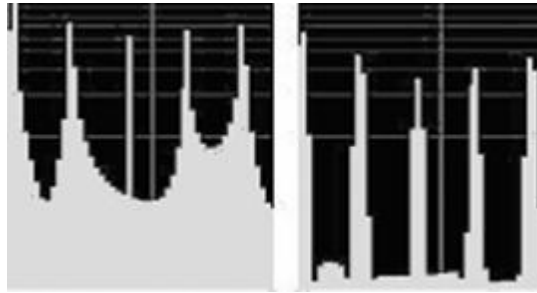


Рисунок 2.6. Дія вікна згладжування Хеммінга

11) Нормування частотного спектру.

Так як усі обчислення в нейронних мережах проводяться над числами з плаваючою точкою, значення параметрів об'єктів, що класифікуються з допомогою нейронних мереж обмежені діапазоном $[0.0, 1.0]$.

12) Логарифмічне стиснення спектра.

Дослідження показали, що інформативність різних частин спектра неоднакова: в низькочастотній області спектра міститься більше інформації, ніж в високочастотній області спектру.

Тому для більш економного використання входів нейронної мережі і збільшення необхідно зменшити число елементів спектру з високою частотою. Це і означає стиснення високочастотної області спектра в просторі частот. В системі SAS застосований найбільш поширений і простий метод – логарифмічне стиснення, або mel-стиснення [17].

13) Застосування вейлвет-перетворень.

У щойно описаній системі SAS для виділення з промови синтаксичних елементів застосовувалося швидке перетворення Фур'є. Однак, Фур'є має свої мінуси та недоліки, в наслідок чого інформація про аналізовані акустичні сигнали може бути втраченою.

Термін вейвлет (wavelets) можна перевести як «маленька хвиля». Вейвлети представляють собою новий інструмент вирішення різних завдань прикладної математики.

Вейвлет-аналіз, детальне знайомство з яким вимагає певних знань в математиці, позбавлений вищезгаданих недоліків. Він дозволяє досягти непоганих результатів при використанні в системах розпізнавання мовлення.

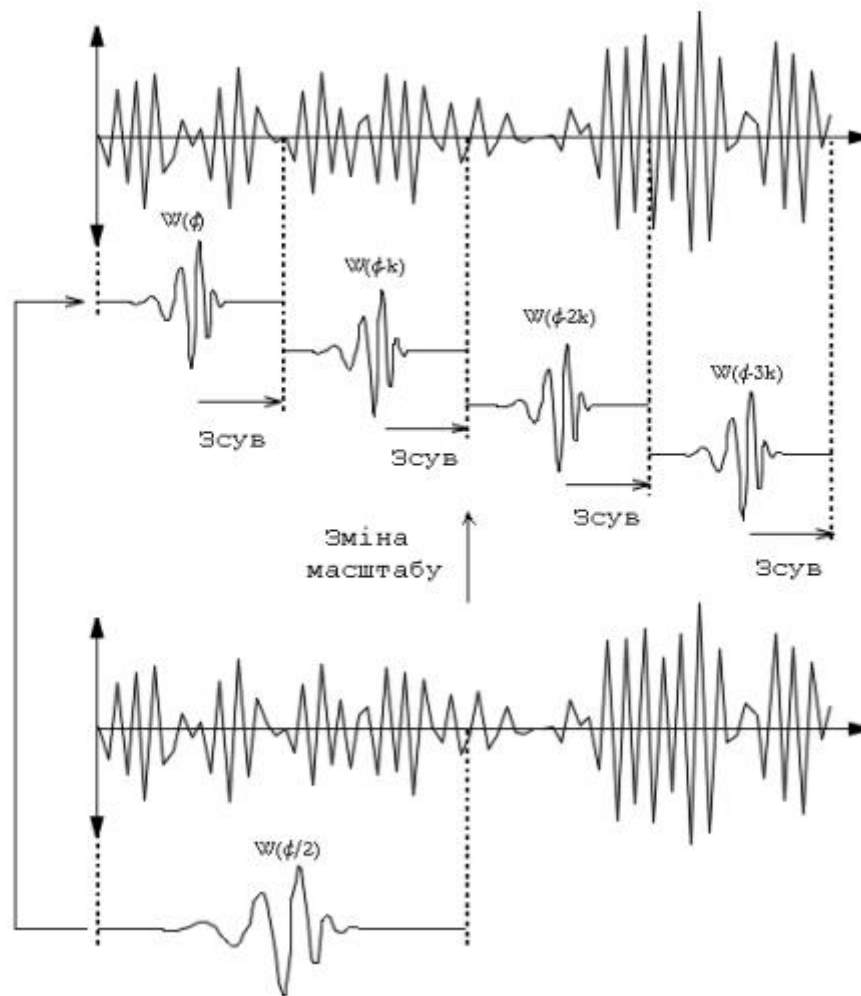


Рисунок 2.7. Представлення сигналу за допомогою зсуву вейвлета

Щоб підкреслити ту обставину, в зарубіжній літературі Фур'є аналіз називають *single spectrum*, а спектри, отримані на основі вейвлет перетворень – *time-scale spectrum*, або *wavelet spectrum*.

Функції-базиси для вейвлетного перетворень побудовані на основі похідних функцій Гаусса.

При застосуванні вейвлет-перетворень для розпізнавання мовлення велику роль грає сам розробник. Адже він повинен обрати оптимальну функцію. Точність системи розпізнавання залежить саме від цього.

Існують два підходи до розпізнавання мовлення. Перший підхід реалізує розпізнавання елементів мовлення за зразком і застосовується в різних роду системах голосового управління. Другий підхід заснований на виділенні в

мовленні лексичних елементів – фонем, алофонів, морфем і т.д. Цей підхід придатний для створення систем диктування тексту [18].

Виділяються системи аналізу акустичних сигналів, що потребують попереднього навчання і залежать від диктора, а також системи, здатні працювати без попереднього навчання і, отже, не залежні від диктора.

Перед тим як приступити до виділення з промови лексичних елементів, необхідно виконати попередню обробку акустичного сигналу. В ході цієї обробки з сигналу видаляються шуми, виконується частотна фільтрація і інтерпретація, а також нормалізація рівня сигналу. Були розглянуті дві методики виділення з промови лексичних елементів.

Перша методика передбачає використання дискретного перетворення Фур'є. При цьому мова представляється у вигляді деякого набору числових параметрів, так як нейронні мережі працюють саме з наборами таких параметрів. Друга методика виділення лексичних елементів мови, заснована на застосуванні вейвлет-перетворень [19].

На відміну від дискретного перетворення Фур'є, цей метод виключає втрату інформації про тимчасових характеристиках оброблюваних сигналів.

Ми відзначили, що при використанні вейвлет-перетворень вхідний сигнал розкладається не в базисі періодичних функцій (як в дискретно перетворенні Фур'є), а в базисі функцій, що характеризують як частоту, так і час.

Метою розпізнавання мовлення є те, що машина може "чути", "зрозуміти" і "діяти" на основі отриманої усної інформації.

Метою автоматичного розпізнавання мовця є аналіз, витяг характеристик та розпізнавання інформації про ідентифікацію мовця. Система розпізнавання особи, що говорить може розглядатися як робота у чотири етапи.

Таблиця 2.1 Методи виокремлення

Метод	Властивості	Реалізація
Аналіз незалежних компонент	Нелінійний метод виокремлення особливостей, лінійне відображення, ітеративний не Гаусівський	Сліпе розділення джерела для не Гаусівський джерел поширення
Кодування з лінійним предиктором	Статичний метод виокремлення особливостей, коефіцієнти нижчого порядку від 10 до 16	Він використовується для функції виокремлення у нижчому порядку
Аналіз основних компонент	Нелінійний метод виокремлення особливостей, лінійне відображення, швидкий, базоване на власному векторі	Традиційно, метод базований на власному векторі. Гарно підходить для даних Гауса
Дискримінантний аналіз	Нелінійний метод виокремлення особливостей, лінійне відображення із спостерігачем, швидкий, базований на власному векторі	Краще, ніж АОК для класифікації
Аналіз кепстру	Статичний метод виокремлення особливостей, спектр потужності	Використовується для представлення спектральної оболонки

Продовження таблиці 2.1

Аналіз частоти	Статичний метод виокремлення особливостей, спектральний аналіз	Спектральний аналіз проводиться з фіксованою роздільною здатністю вздовж суб'єктивної шкали частот, тобто мель-частотної шкали.
Блок фільтрів	Масив смугових фільтрів, які розділяють вхідний сигнал на кілька компонент, кожна з яких несе певну частоту піддіапазону	Розділяє вхідний сигнал на кілька компонент
Кепстер частоти мелів	Спектр потужності обчислюється за допомогою аналізу Фур'є	Метод використовується для виокремлення особливостей
Метод виокремлення особливостей, оснований на ядрі	Нелінійна трансформація	Зменшення розмірності призводить до кращої класифікації, і це використовується для надлишкових особливостей, а також покращення класифікації помилок

Закінчення таблиці 2.1

Вейвлет	Кращий час рішення для перетворення Фур'є	Замінює фіксовану пропускову здатність перетворення Фур'є на пропорційну частоту, що забезпечує кращий час виконання на високих частотах, ніж перетворення Фур'є
Динамічне виокремлення особливостей	Коефіцієнти прискорення та дельти, тобто похідні 2 та 3 порядку звичайних коефіцієнтів MFCCs	Використовується для динамічних особливостей
Спектральне видалення	Надійний метод виділення особливостей	Його використання базується на спектрограмі
Видалення середнього кепстру	Надійний метод виділення особливостей	Схожий на MFCCs, але працює на середніх статистичних параметрах
RASTA фільтрування	Для зашумленої мови	Виокремлює особливості згід зашумлених даних
Інтегрований метод фонемного підпростору	Трансформація, основана на АОК, дискримінантному аналізі та аналізі незалежних компонент	Найбільша точність з існуючих методів

Як бачимо, є різні методи виокремлення особливостей, які мають певні відмінності в реалізації і використовуються для різних задач. Усі вони знаходяться в процесі вдосконалення, деякі з'являються як доопрацювання або поєднання кількох інших методів. Після завершення етапу виокремлення особливостей ми можемо переходити до етапу моделювання [20].

Мовленнєва інформація містить різні типи інформації, що відображають особистість мовця. Це інформацію про спікер, пов'язану з голосовим трактом, джерелом збудження та особливостями поведінки. Останній тип інформації також вбудований в сигнал і його можна використовувати для розпізнавання особи, що говорить. Етап аналізу мови працює з рівнями з підходящим фреймом для сегментації сигналу для його подальшого аналізу та виділення. Метод аналізу мовлення проводиться за допомогою трьох технік: аналіз сегментів, аналіз підсегментів, та надсегментарний аналіз. Під час аналізу сегментів використовується розмір фрейму та зсув в області 10-30 мілісекунд для виявлення інформації про мовця, а саме для вилучення інформації про голосовий тракт. Аналіз підсегментів використовує розмір фрейму та зсуву в межах 3-5 мс. і використовується, в основному для аналізу та виокремлення інформації про стадії збудження. Надсегментарний аналіз базується на розпізнаванні характеристик, пов'язаних з особливістю відношень між окремими звуками у мовця.

Проблема виокремлення особливостей в задачі класифікації полягає у зменшенні розмірності вхідного вектора при збереженні дискримінаційної потужності сигналу. Як ми знаємо з фундаментального формування системи ідентифікації та перевірки динаміків, кількість навчально-тестових векторів, необхідних для класифікації, зростає з розмірами даного вводу, тому нам потрібна функція виокремлення мовного сигналу.

Мета методу моделювання полягає в тому, щоб генерувати моделі мовця з використанням вектора специфічних особливостей спікера. Техніка моделювання мовця розділена на дві класифікації – розпізнавання спікера та ідентифікація спікера. Метод ідентифікації мовця автоматично визначає, хто

говорить на основі індивідуальної інформації, інтегрованої в мовний сигнал. Розпізнавання мовця також ділиться на дві частини, та, що залежить від спікера і не залежить від нього. У режимі, незалежному від мовця, комп'ютер повинен ігнорувати специфічні характеристики мовного сигналу спікера та витягувати передбачене повідомлення. З іншого боку, в режимі, залежному від спікера, машина повинна витягувати характеристики особи, що говорить в акустичному сигналі.

Головною задачею розпізнавання мовця є порівняння мовного сигналу невідомої особи з базою даних уже відомих осіб. Розпізнавання мовців також можна розділити на два методи, текстові та незалежні від тексту методи. За допомогою текстового методу оратор говорить ключові слова або речення, що мають один і той самий текст як для навчання, так і для розпізнавання. В той час, коли інший метод не покладається на конкретні тексти у мові.

Висновки до розділу

Алгоритм роботи системи робота-помічника повинен описувати повний цикл взаємодії з користувачем. Починаючи від того в якому режимі знаходиться робот в стані спокою, в нашому випадку це буде режим очікування кодового слова як сигнал про початок обробки голосу та порівняння отриманого з базою даних типових команд, закінчуючи тим що повинен бути прописаний алгоритм поведінки в ситуаціях, коли система не змогла розібрати слово, або не знайшла відповідної команди.

Алгоритм має бути циклічним, адже після виконання роботом поставленої задачі він повинен повертатися до режиму очікування. Алгоритм також має бути стійким до помилок, адже існує багато сценаріїв в такій складній системі як «розумний дім», коли неправильна інтерпретація голосових команд може призвести до небажаних наслідків або до тупикових та невизначених ситуацій. Комунікація повинна бути також інтуїтивно зрозумілою для користувача.

Система повинна враховувати всі можливі помилки та правильно реагувати. Сповіщати користувача в момент переходів з різних станів, щоб уникати помилок. Також команди занесені до бази даних повинні значно відрізнятися між собою, щоб уникати небажаних спрацювань.

3. ОПИС РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Вибір середовища та засобів розробки

Розробка системи ведеться у програмному середовищі системної розробки PyCharm від компанії JetBrains, розробника не менш популярного середовища для розробки мовою Java, IntelliJ IDEA.

Основними засобами розробки можна назвати наступне:

1. Мова програмування Python 3
2. Серверний мікрофреймворк Flask
3. Реляційна база даних PostgreSQL
4. Мова програмування та засоби JavaScript, HTML, CSS
5. Фреймворк SpeechRecognition
6. Фреймворк pyttsx
7. Фреймворк PyAudio

3.1.1. Вибір мови програмування

Мова програмування Python обрана як найкращий варіант для реалізації даної системи через низку причин. Найголовнішою є те, що Python – інтерпретована мова, тобто вона не компілюється, а послідовно виконується, як є, що дає їй змогу бути незалежною від платформи, де програми, створені її засоби, запускаються.

Це також дає можливість до рефлексії, тобто структура і поведінка компонентів програми за час виконання можуть змінюватися довільним чином, що вигідно нам як розробникам. Наостанок, через інтерпретованість, програми на Python важать суттєво менше, ніж програми на компільованих мовах.

Мова Python – це інтерпретована високорівнева мова програмування загального призначення. Створив її Гвідо ван Россум, випустивши вперше у 1991 році, маючи в основі філософію, спрямовану на читабельність коду, зокрема шляхом екстенсивного використання пробілів. Вона надає

розробнику доступ до конструкцій, що забезпечують прозоре програмування як у малих, так і великих масштабах.

Python володіє динамічною системою типів та автоматичним керуванням пам'яттю. Вона підтримує велику кількість парадигм програмування, включаючи об'єктно-орієнтоване, імперативне, функціональне та процедурне, та має потужну і вичерпну стандартну бібліотеку.

Інтерпретатори мови Python доступні для великої кількості операційних систем. Зокрема, CPython, еталонна реалізація мови Python, є відкритою системою з суспільною моделлю розробки, як і більшість її варіативних імплементацій. Вона оперується Фондом Програмного Забезпечення Python.

Основну концепцію мови було формалізовано в кінці 1980-х років, а перші реалізації почалися у грудні 1989 року за ініціативи Гвідо ван Россума у Національному Дослідницькому Інституті Математики та Комп'ютерних Наук у Нідерландах. Вона є наслідником мови ABC (що, в свою чергу, є наслідником SETL), що мала здатність до обробки виключень та легкої взаємодії з операційною системою Атомеба. Гвідо ван Россум залишається головним автором мови [21].

16 жовтня 2000 року було випущено Python 2.0, що мала велику кількість нових можливостей, зокрема збиральника сміття з виявленням циклів та підтримку символів Unicode. Після цього релізу, розробка стала більш прозорою та підкріпленою спільнотою.

3 грудня 2008 року вийшов Python 3, після довгого періоду тестування. Ця версія уявляє собою велику ревізію всієї мови, що не має повної підтримки минулих версій. Однак, велика кількість головних можливостей були портовані на старіші версії, 2.6.x та 2.7.x, а також було випущено утиліту 2to3, що автоматично перекладає код Python 2 на Python 3.

Python є багатопарадигмальною мовою програмування. Повністю підтримуються об'єктно-орієнтоване та структурне програмування, а багато можливостей мови підтримують функціональні та аспектно-орієнтовані стилі

(включаючи метапрограмування та метаоб'єкти – магічні методи). Багато інших парадигм підтримуються за допомогою розширень, зокрема логічне програмування.

Python підтримує динамічну типізацію, а також поєднує можливість рахування посилань та збиральника сміття з виявленням циклів для керування пам'яттю. Також вона підтримує динамічне розширення імен, що пов'язує імена методів та змінних лише протягом виконання програми.

Архітектура мови підтримує деякі можливості функціонального програмування у стилі мови Lisp. Наявні методи `filter()`, `map()`, `reduce()`, спискові вирази, словники та набори, а також генератори виразів. Стандартна бібліотека містить два пакети (`itertools` та `functools`), що впроваджують функціональні інструменти, запозичені з мов Haskell та Standard ML.

Також Python дуже зручний як платформа для запуску легковісних серверів та сервісів, побудови мікросервісних архітектур тощо. Це і дає нам змогу так легко і швидко розгорнути нашу систему і забезпечити її модульність. Також, для Python існує дуже велика кількість бібліотек, доступних будь-кому, що дозволяють виконувати найрізноманітніші операції, користуватися великою кількістю сервісів без необхідності звертатися напяму до API того чи іншого сервісу, а також виконувати складні обчислення чи операції, пов'язані зі штучним інтелектом. Це все робить Python нашою мовою вибору для цієї системи.

Зокрема, ми використовуємо серверний фреймворк Flask, написаний для мови Python. Він є дуже легким і простим у користуванні, має велику кількість завантажуваних бібліотек для реалізації необхідних елементів веб-сервісу, як то авторизація чи робота з базою даних. На його основі працюватиме головний модуль всієї системи.

Flask – це веб-мікрофреймворк, написаний на мові Python. Його називають мікрофреймворком через те, що він не вимагає жодних конкретних інструментів чи бібліотек для роботи. У нього не має рівню абстрації для баз даних, валідацій форм, чи будь-яких інших компонентів, функціонал яких

забезпечується, зазвичай, готовими сторонніми бібліотеками. Втім, Flask, підтримує велику кількість розширень, що додають можливості додатку так, ніби вони були реалізовані безпосередньо у самому мікрофреймворку.

Також існують розширення для об'єктно-реляційного мепінгу, валідації форм, обробки завантажень, відкритих технологій аутентифікації та деяких інструментів, що загалом стосуються фреймворків. Розширення оновлюються набагато частіше, ніж саме ядро програми Flask. Із проектів, що використовують Flask, можна відзначити Pinterest, LinkedIn та саму сторінку фреймворку.

Двома основними компонентами мікрофреймворку Flask є Werkzeug та Jinja2.

Werkzeug – це бібліотека для мови програмування Python, що уявляє собою набір інструментів для реалізації WSGI-додатків, і поширюється за ліцензією BSD. Werkzeug дає можливість створювати об'єкти для обробки запитів, відповідей та інших функцій. Його можна використати для побудови власного програмного фреймворку у вигляді надбудови, і він підтримує як Python 2, так і Python 3.

Jinja2 – двигун роботи з шаблонами для мови програмування Python, що схожий на веб-фреймворк Django, де шаблони обчислюються у окремому середовищі.

Flask підтримує такі можливості, як наявність серверу розробки та дебаггеру, вбудованих модулів для unit-тестів, обробки RESTful-запитів, захищених cookies, а також має вичерпну документацію.

```
from flask import Flask, escape, request

app = Flask(__name__)

@app.route('/')
def hello():
    name = request.args.get("name", "World")
    return f'Hello, {escape(name)}!'
```

Рисунок 3.1. Підключення Flask до проекту

3.1.2. Вибір бази даних

Із можливих до застосування варіантів баз даних можна назвати:

1. MySQL.
2. PostgreSQL.
3. SQLite.
4. MongoDB.

MySQL – перевірена часом і стійко стабілізована реляційна база даних, зручна для зберігання великої кількості даних, що пов'язані між собою певними типами зв'язків. До того ж типу відносяться PostgreSQL та SQLite, однак перша з них має велику кількість додаткових типів даних, можливостей і надстройок, а друга уявляє собою полегшену і мобільну версію MySQL. Останній же варіант є прикладом так званих NoSQL-баз даних, які засновані на методі зберігання даних { ключ: значення }. Вони є зручними для зберігання великої кількості даних, що між собою пов'язані можуть бути слабо, однак виграє у швидкості на голову.

PostgreSQL – об'єктно-реляційна база даних, що фокусується на розширюваності та відповідності стандартам. Її основні функції, як сервера бази даних, є надійне зберігання даних та повертання їх у відповідь на запити від інших програмних додатків.

Вона може працювати із навантаженнями, що варіюються від малих додатків на одній машині до масивних додатків із виходом до мережі Інтернет та великою кількістю одночасних користувачів. Вона є базою даних за замовчуванням у операційних системах компанії Apple, але також доступна і для операційних систем Microsoft Windows та Linux.

PostgreSQL є транзакційною, і підтримує основні принципи транзакцій баз даних, такі як атомарність, постійність, ізолюваність та стійкість. Вона містить у собі функціонал оновлюваних та матеріалізованих представлень даних, трігерів, зовнішніх ключів, а також підтримує функції та збережувані процедури.

Це є безкоштовна система з відкритим кодом, що розробляється широкою групою компаній та окремих вкладників, і поширюється за ліцензією PostgreSQL.

Вона підтримує такі можливості зберігання та маніпуляції даними, як реплікації, індексування, створення схем, має розширену кількість типів даних, об'єктів, що визначаються користувачами, наслідування, інтерфейсів, тригерів тощо.

3.1.3. Вибір бібліотеки та інструментів розпізнавання мовлення

SpeechRecognition – бібліотека, написана мовою Python, що забезпечує інструменти для розпізнавання мовлення, та підтримує декілька двигунів та API. Вона підтримує версії Python 2 та 3, а також вимагає для роботи бібліотеку PyAudio. Це бібліотека, що надає інструментарій для роботи з пристроями мікрофонів, тобто запису і зберігання аудіоінформації, а також подання її у вигляді потоку.

Pytsx – кросс-платформенна обгортка для перетворення тексту на мову, написана мовою Python. Вона володіє дуже простим інтерфейсом взаємодії, а також підтримує декілька двигунів відтворення голосу для різних платформ.

Варто відзначити також і те, що для розробки ми використовуємо середовище PyCharm, який розроблений спеціально для роботи з цією мовою програмування, має інтегровану підтримку більшості фреймворків, вбудований термінал, консоль мови Python, дуже зручний дебагер, що скорочує процес прибирання багів у багато разів, та підтримку системи контролю версій (git тощо).

3.2. Програмні модулі

Загалом, всю систему можна поділити на наступні модулі:

1. Модуль веб-сервера інтерфейсу взаємодії.
2. Модуль запису, обробки, інтерпретації та синтезу мови.
3. Додаткові скрипти.

3.2.1. Модуль розпізнавання мовлення

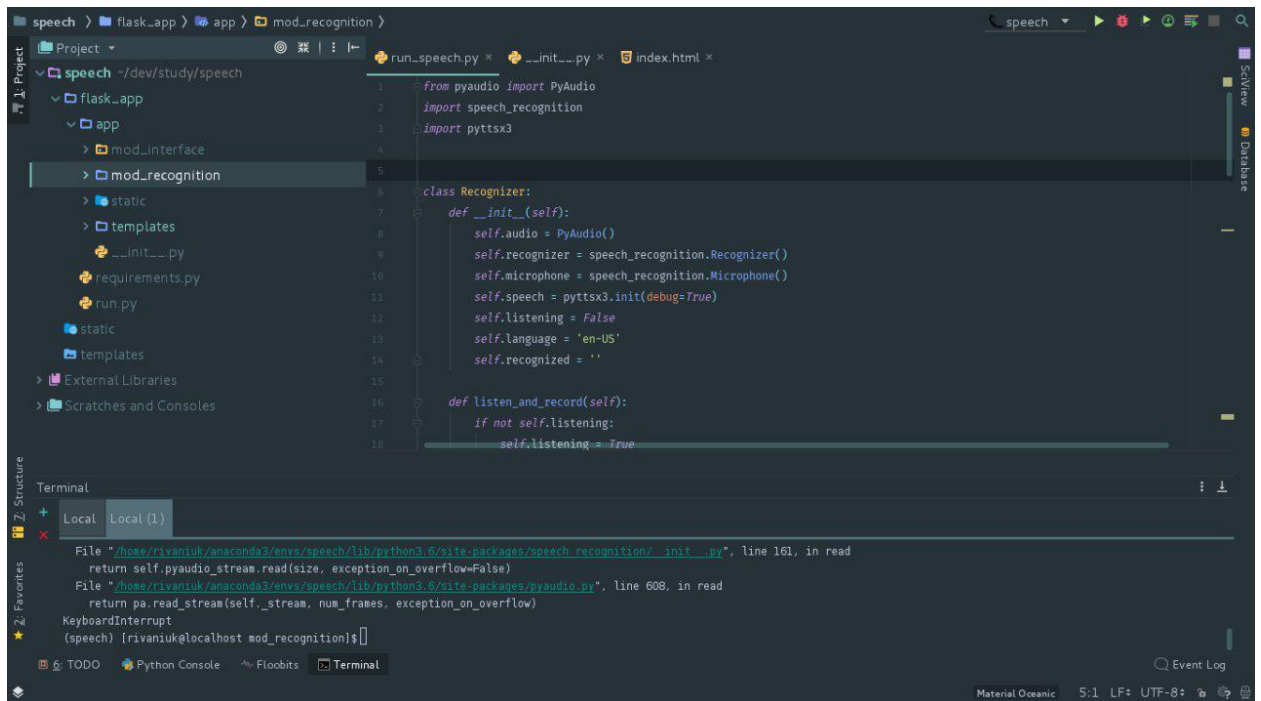


Рисунок 3.2. Зовнішній вигляд середовища розробки

Модуль запису і обробки мови має такі можливості:

1. Циклічний запис інформації за допомогою мікрофона.
2. Обробка записаної інформації та перетворення на текст.
3. Зберігання результатів обробки у файлах системи.
4. Відтворення голосу у відповідь на команду.

Основною перевагою системи є її модульність та монолітність: веб-інтерфейс реалізовано окремим модулем, який отримує інформацію від окремого незалежного циклічного процесу, що запускається паралельно.

Розгортатися це все буде на локальному хості, де знаходитиметься сама система. Відповідна адреса сервіса – 0.0.0.0:5000.

Якщо технічна частина розташована безпосередньо у розумному домі і має достатню потужність і оснащена достатньо потужною системою Wi-Fi приймання та передачі, додаткового розгортання для сервісів не потрібно. В іншому випадку, можна застосувати засоби хмарного деплою, як-то систему Azure від Microsoft, AWS від Amazon тощо.

Як видно з рис. 3.3, структуру ми маємо наступну. У головній директорії проекту знаходяться модуль самого додатку.

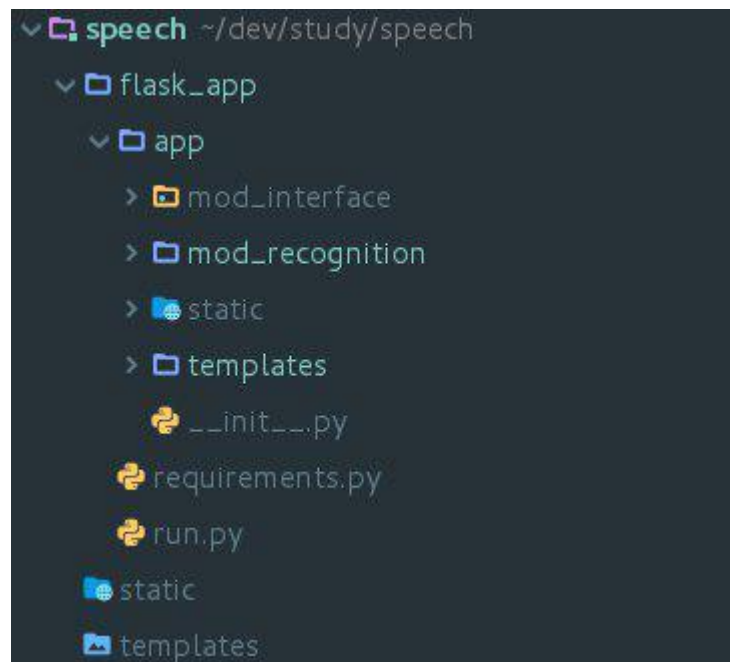


Рисунок 3.3. Структура проекту

Всередині модуля додатку знаходяться підмодулі веб-інтерфейсу та розпізнавання голосу. У першому реалізована логіка контролерів відображення веб-сторінок, а також взаємодії з клієнтом та файлом результатів розпізнавання, а ще директорії зі статичними файлами CSS та JS, а також директорія із HTML-шаблоном. У другому знаходиться реалізація підмодуля розпізнавання та відтворення голосу, а також директорія для файлів результатів. Також наявні додаткові скрипти для встановлення необхідних залежностей та компонентів.

PyAudio - основний об'єкт модуля, який представляє можливість роботи із звуком.

Створено екземпляр класу розпізнавача голосу, який міститиме екземпляри класів аудіодрайвера, розпізнавача голосу, мікрофона та відтворювача голосу, а також стану прослуховування, поточну мову і останній розпізнаний результат.

```

class Recognizer:
    def __init__(self):
        self.audio = PyAudio()
        self.recognizer = speech_recognition.Recognizer()
        self.microphone = speech_recognition.Microphone()
        self.speech = pyttsx3.init(debug=True)
        self.listening = False
        self.language = 'en-US'
        self.recognized = ''

    def listen_and_record(self):
        if not self.listening:
            self.listening = True
            with self.microphone as mic:
                while True:
                    try:
                        print('Adjusting')
                        self.recognizer.adjust_for_ambient_noise(mic)
                        print('Listening')
                        res = self.recognizer.listen(mic, timeout=3, phrase_time_limit=3)
                        if res:
                            print('Recognizing')
                            recognized = self.recognizer.recognize_google(res,
                                                                           language=self.language)
                            if recognized:
                                self.recognized = recognized

```

Рисунок 3.4. Частина коду розпізнавача

3.2.2. Модуль користувацького інтерфейсу

До можливостей модуля графічного користувацького інтерфейсу можна віднести:

1. Спостерігання поточного стану роботи.
2. Надавання йому команд засобами окрім голосового вводу.
3. Перезапуск системи, робота.
4. Спостерігання процесу розпізнавання в реальному часі.

Для розробки модуля користувацького інтерфейсу обрано JavaScript бібліотеку React.

React – відкрита JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту веб-сторінки, з якими стикаються в розробці односторінкових застосунків. Розробляється Facebook, Instagram і спільнотою індивідуальних розробників.

React дозволяє розробникам створювати великі веб-застосунки, які використовують дані, котрі змінюються з часом, без перезавантаження сторінки. Його мета полягає в тому, щоб бути швидким, простим, масштабованим. React обробляє тільки користувацький інтерфейс у застосунках. Це відповідає видові у шаблоні модель-вид-контролер (MVC), і може бути використане у поєднанні з іншими JavaScript бібліотеками або в великих фреймворках MVC, таких як AngularJS. Він також може бути використаний з React на основі надбудов, щоб піклуватися про частини без користувацького інтерфейсу побудови веб-застосунків. Як бібліотеку інтерфейсу користувача React найчастіше використовують разом з іншими бібліотеками, такими як Redux.

Його перевагами є:

- декларативність. React спрощує створення інтерактивних інтерфейсів. Вам потрібно лише описати, як різні частини інтерфейсу виглядають у кожному стані вашого додатку і React ефективно оновить та відрендерить лише потрібні компоненти, коли ваші дані зміняться. Декларативні інтерфейси роблять ваш код більш передбачуваним і його набагато легше налагоджувати;
- заснований на компонентах. Дає змогу створювати інкапсульовані компоненти, які керують власним станом, а з них будувати складні інтерфейси. Оскільки логіка компонентів написана на JavaScript, замість шаблонів, ми з легкістю можемо передавати складні дані у нашому додатку і зберігати стан окремо від DOM;
- є бібліотекою. Дає змогу до гнучкості у розробці;

Середовищем розробки користувацького модуля обрано Visual Studio Code. Це засіб для створення та редагування сучасних веб-застосунків і програм для хмарних систем. Visual Studio Code розповсюджується безкоштовно і доступний у версіях для платформ Windows, Linux і OS X.

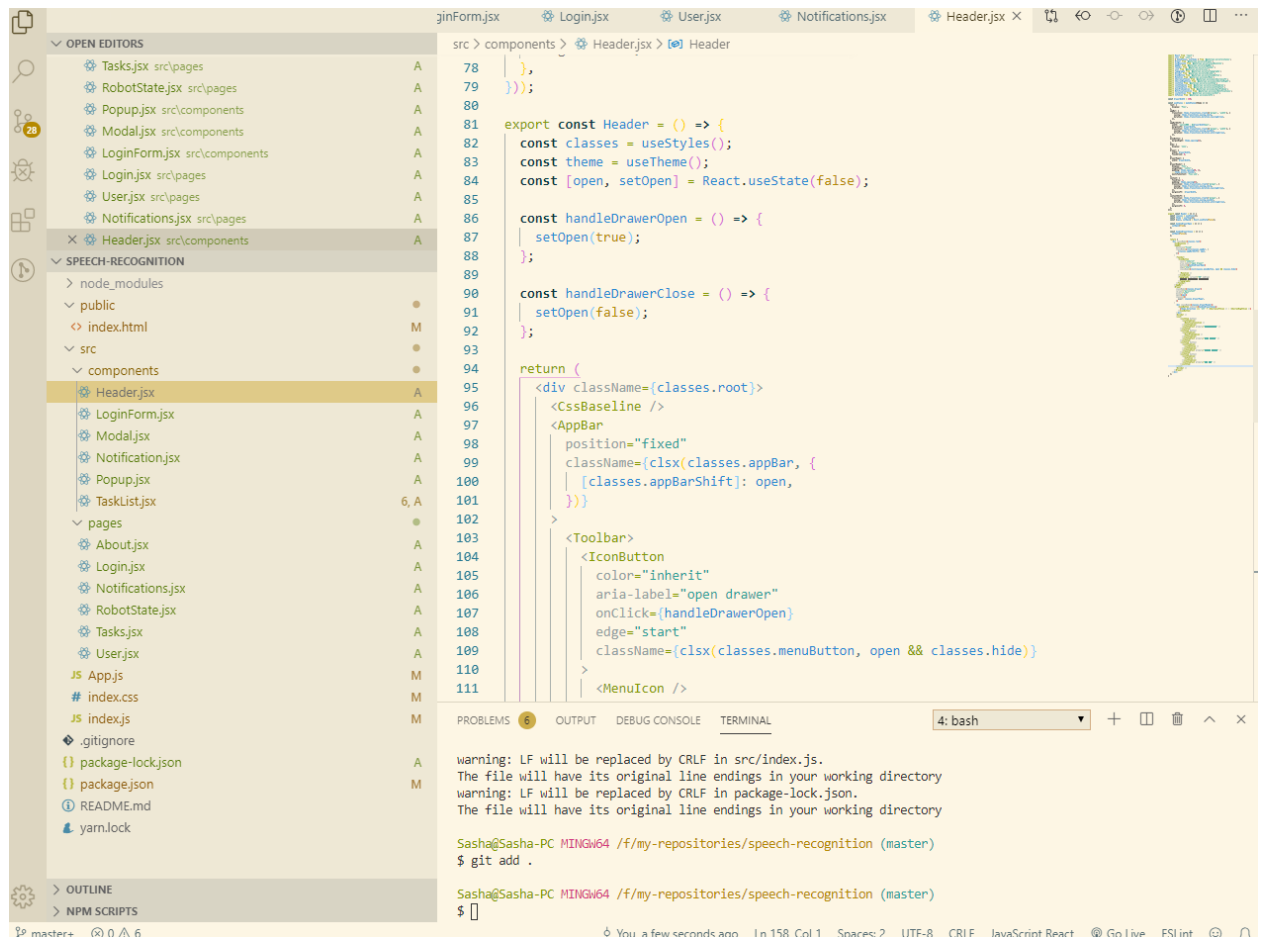


Рисунок 3.5. Скріншот середовища розробки користувацького інтерфейсу

Виходячи з поставлених задач до функціоналу додатку, а саме можливих функцій побудуємо навігаційне меню.

Воно буде включати в себе такі роути:

- повідомлення користувача про зміни стану роботи, виконання задач;
- поточний стан робота;
- список наявних команд
- інформаційний розділ про додаток.

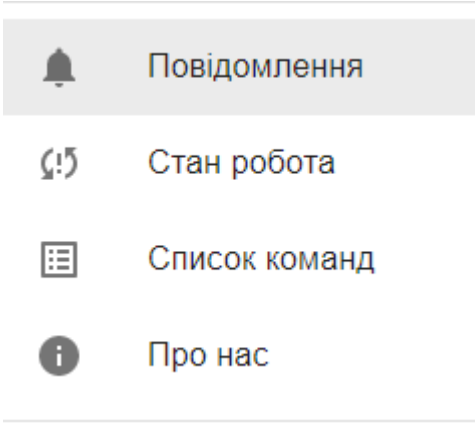


Рисунок 3.6. Навігаційне меню

Для стилів компонентів використано фреймворк Material UI. Структура проекту буде складатися з компонентів та сторінок.

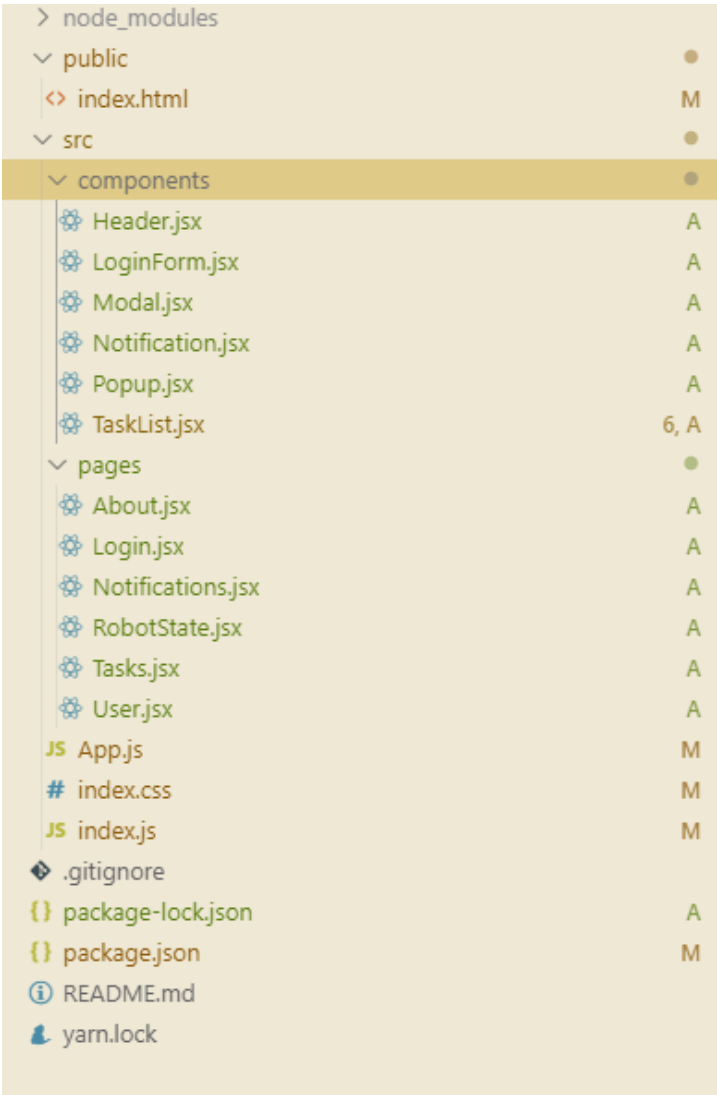


Рисунок 3.7 Структура проекту користувацького інтерфейсу

Material UI – бібліотека стилізованих React компонентів від Google, що значно збільшує швидкість розробки. При цьому бібліотека залишає можливість кастомізувати усі компоненти в залежності від поставленої задачі розробки користувацьких інтерфейсів.

Для контролю стану додатку обрано бібліотеку Redux. Це значно покращить процес розробки та контроль стану компонентів.

```
const reducer = (state = initialState, action) => {
  switch (action.type) {
    case 'USER_LOGGED_IN':
      const user = state.users.find(user => user.email === action.email);

      if (user && user.password !== action.password) {
        return {
          ...state,
          invalidEmail: false,
          invalidPassword: true
        };
      }

      if (user && user.password === action.password) {
        return {
          ...state,
          loggedIn: true,
          currentUser: user,
          invalidEmail: false,
          invalidPassword: false
        };
      }

      return {
        ...state,
        invalidEmail: true
      };
  }
};
```

Рисунок 3.8. Логіка редакс для авторизації користувача

Redux – відкрита JavaScript бібліотека призначена для управління станом програми. Найчастіше використовується разом з React або Angular для побудови інтерфейсів користувача.

Redux був створений Деном Абрамовим і Ендрю Кларком у 2015 році. Redux був створений під впливом Flux, розробленого Facebook, і під впливом функційної мови програмування Elm.

Redux – це контейнер станів для застосунків JavaScript. Він допомагає розробникам оптимізувати код програми. Крім того, він забезпечує вдосконалення досвіду розробника, наприклад, редагування живого коду в поєднанні з відладчиком, що працює під час роботи.

Redux можна використовувати разом із React або іншими бібліотеками. Розмір файлу Redux невеликий, 2kB, включаючи залежності.

Redux зберігає стан всього застосунку в дереві об'єктів в одному сховищі. Одне дерево станів полегшує налагодження або перевірку програми; це також дозволяє зберігати стан вашого застосунку в процесі розробки, для прискорення циклу розробки.

Єдиний спосіб змінити стан – це виокремити дію, об'єкт, що описує те, що сталося. Це гарантує, що ні перегляди, ні зворотні виклики мережі ніколи не будуть змінювати стан. Натомість вони виражають тільки намір це зробити. Всі зміни – централізовані і відбуваються одна за іншою у чіткій послідовності. Оскільки дії є простими об'єктами, вони можуть бути зареєстровані, серіалізовані, збережені та пізніше відтворені для налагодження або тестування.

Редьюсери – це лише чисті функції, які приймають попередній стан і дію, і повертають наступний стан. В процесі розробки редьюсери можуть бути розділені на дрібніші редьюсери, які управляють певними частинами дерева станів. Оскільки редьюсери - це лише функції, ви можете контролювати порядок їх надсилання, передавати додаткові дані або навіть створювати повторювані редьюсери для звичайних завдань, таких як розбиття на сторінки.

Додаток повинен вміти отримувати дані з бази даних для переліку можливих команд. Команди можуть викликатися з пристрою, а не тільки за допомогою голосу. Модуль взаємодіє з іншими модулями для отримання інформації про стан роботи.

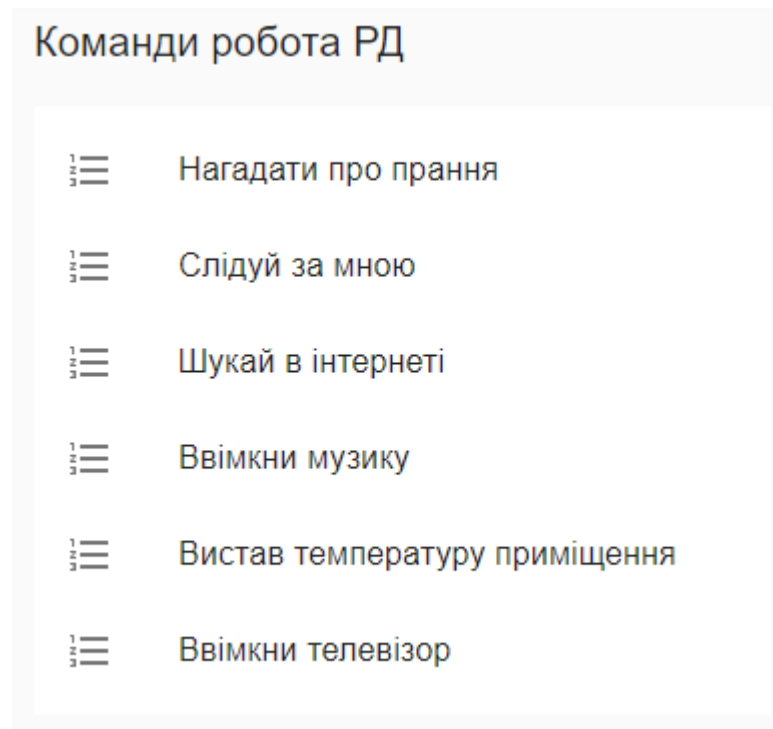


Рисунок 3.9. Вікно з переліком можливих команд

Представлено лише невелику кількість команд, деякі з яких мають підзадачі. Наприклад разом з «шукай в інтернеті» система очікує почути пошуковий запит.

Висновки до розділу

У цьому розділі ми представили та пояснили технології, які застосовуються у цьому проекті, а також безпосередню виконання нашої системи. В решті, ми отримали програмну реалізацію системи, що здатна у реальному часі прослуховувати та розпізнавати голосову інформацію, що надається користувачем-господарем, інформувати господаря у відповідь та передавати відповідні сигнали далі для виконання іншими можливими модулями чи пристроями.

Засоби і бібліотеки, що ми застосували для розробки, дозволили суттєво спростити її, полегшити майбутню підтримку, додавання нових функцій, вбудовуваність у готові технічні комплекси. За допомогою веб-інтерфейсу ми можемо споглядати у реальному часі інформацію про стан роботи та системи,

який у свою чергу забезпечується архітектурою, що ми застосували під час розробки.

Всі використані технології є сучасними та їх підтримка ведеться великими компаніями. Це дає впевненість у тому, що технології будуть продовжувати своє існування та будуть актуальними. Це забезпечить працездатність модулів нашого програмного забезпечення впродовж тривалого часу.

4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ

Процес розробки та виведення стартап-проекту на ринок має містити виконання деяких кроків, в яких визначаються ринкові перспективи проекту, графік та принципи організації виробництва, фінансовий аналіз та аналіз ризиків і заходи з просування пропозиції для інвесторів.

Метою розділу є формування інноваційного мислення, підприємницького духу та формування здатностей щодо оцінювання ринкових перспектив і можливостей комерціалізації основних науково-технічних розробок, сформованих у попередній частині магістерської дисертації у вигляді розроблення концепції стартап-проекту в умовах висококонкурентної ринкової економіки глобалізаційних процесів.

В загальному, етапи розробки стартап-проекту можна представити наступним чином:

1) Маркетинговий аналіз стартап-проекту.

В межах цього етапу:

- відбувається опис ідеї проекту та визначаються загальні напрямки використання потенційного товару чи послуги, а також їх відмінність від конкурентів;
- аналізуються можливості ринку щодо реалізації проекту;
- на основі аналізу середовища ринку розробляється стратегія ринкового впровадження потенційного товару в межах проекту.

2) Організація стартап-проекту.

В межах даного етапу:

- розробляється календарний план-графік реалізації проекту;
- розраховується необхідні основні засоби та нематеріальні активи;
- розраховується запланований обсяг виробництва потенційного товару, на основі чого розраховується потреба у матеріальних ресурсах та персоналі;

- вираховуються загальні початкові витрати для запуску стартап-проекту та заплановані загальногосподарські витрати, необхідні для реалізації стартап-проекту.

3) Фінансово-економічний аналіз та оцінка ризиків проекту.

В межах цього етапу:

- вираховується обсяг інвестиційних затрат;
- вираховуються основні фінансово-економічні показники стартап-проекту (обсяг виробництва, собівартість виробництва продукції, ціна реалізації, податкове навантаження, чистий прибуток) та визначаються показники інвестиційної привабливості проекту (запас фінансової міцності, рентабельність продажів та інвестицій, період окупності стартап-проекту);
- вираховується рівень ризикованості стартап-проекту, визначаються головні ризики стартап-проекту та шляхи їх уникнення (реагування на ризики).

4) Заходи з комерціалізації проекту

Цей етап направлений на пошук інвесторів та просування інвестиційної пропозиції (оферти) проекту.

Він передбачає:

- визначення цільових груп інвесторів та опис їх ділових інтересів;
- складання інвестиційної пропозиції (оферти), тобто стислої характеристики стартап-проекту для попереднього ознайомлення інвесторів;
- планування заходів щодо просування оферти: визначення каналів комунікації, планування заходів з просування в рамках обраних каналів;
- планування ресурсів для реалізації заходів щодо просування оферти.

Визначені етапи, реалізовані вчасно та послідовно створюють передумови для успішного старту проекту на ринку.

4.1.Опис ідеї проекту

В рамках підпункту проаналізовано і подано у вигляді таблиць:

- зміст ідеї (що пропонується);
- можливі напрямки застосування;
- основні вигоди, які може отримати користувач товару (за кожним напрямком застосування);
- чим товар відрізняється від існуючих аналогів або замінників.

Зміст ідеї проекту, що дає в цілому картину запропонованого рішення наведено у табл. 4.1.

Таблиця 4.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка програмного забезпечення для голосового управління роботом РД	1. Голосове управління роботом 2. Розпізнавання мовлення	1. Зручний інтуїтивно простий інтерфейс для голосового управління 2. Стійка до помилок система 3. Можливість управління розумним домом за допомогою голосу

Аналіз потенційних техніко-економічних переваг проекту (чим відрізняється від існуючих аналогів та замінників) у порівнянні з конкурентними пропозиціями передбачає:

- визначення набору техніко-економічних властивостей та характеристик проекту;
- визначення попереднього набору конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та збір інформації щодо значень техніко-економічних показників для ідеї проекту та конкурентних проектів відповідно до зазначеного вище переліку;
- проведення порівняльного аналізу показників.

Для майбутнього проекту визначені показники, що мають:

- а) гірші показники (W, слабкі);

- б) аналогічні показники (N, нейтральні);
в) кращі показники (S, сильні) (табл.5.2).

Таблиця 4.2. Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№	Техніко-економічні характеристики ідеї	Продукція конкурентів		W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Конкуренти			
1	Формат сервісу	Web-App	Web-App		+	
2	Кросплатформеність	+	+		+	
3	Вартість генерації	Безкоштовно	Безкоштовно		+	
4	Налаштування	-	+			+
5	Унікальність	+	+		+	
6	Кількість генерацій	Безлімітна	Зменшується з кількістю	+		

Зазначений вище перелік сильних, слабких та нейтральних характеристик потенційного проекту є основою для формулювання його конкурентоспроможності.

4.2. Технологічний аудит ідеї проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару).

Визначення того, чи можливе здійснення ідеї проекту з технологічної точки зору, тобто визначення технологічної здійсненності проекту, передбачає аналіз таких складових (табл.4.3):

- технологія виготовлення товару згідно проекту;

- чи існують технології для виготовлення товару;
- доступність технологій виготовлення авторам проекту.

Таблиця 4.3. Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
1	WEB-додаток	HTML, CSS, JavaScript, React	+	+
2	Модуль розпізнавання мовлення	PostgreSQL, Python,	+	+

За результатами аналізу таблиці робиться висновок щодо можливості технологічної реалізації проекту: так чи ні, а також технологічного шляху, яким це доцільно зробити (з поміж названих технологій обираються такі, що доступні авторам проекту та є наявними на ринку).

Висновок: реалізація продукту з технологічної точки зору можлива.

4.3. Аналіз ринкових можливостей

Визначення можливостей ринку, які можна використати під час впровадження проекту на ринку. Аналіз ринкових загроз, які можуть перешкодити успішній реалізації стартап-проекту, що дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів. Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку табл.4.4.

Таблиця 4.4. Попередня характеристика потенційного ринку

№	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	6
2	Загальний обсяг продаж, грн./ум.од	1000000
3	Динаміка ринку	зростає
4	Наявність обмежень для входу	відсутні
5	Специфічні вимоги до стандартизації та сертифікації	відсутні
6	Середня норма рентабельності в галузі або по ринку, %	R=30%

Висновок: враховуючи яка існує кількість головних гравців на ринку, а також зростаючу динаміку ринку та середню норму рентабельності є можливість зробити висновок, що ринок є привабливим для реалізації проекту.

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці цільових груп клієнтів	Вимоги споживачів до товару
1	Потреба в інтерфейсах голосового управління	Власники розумних будинків, компанії розробники	Потребують високої точності розпізнавання. Потребують повний контроль над системою.	Повинна бути зручною, інтуїтивно зрозумілою

Після проведення аналізу потенційних груп клієнтів визначено умови ринкового середовища. Для цього створено таблиці з факторами, що сприяють впровадженню проекту на ринок, та факторів, що цьому перешкоджають (табл.4.6, табл.4.5).

Таблиця 4.6. Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Наявність конкурентів котрі пропонують подібні рішення	1. Розробка унікальних характеристик товару; 2. Збільшення якості товару
2	Інвестиції	Недостатнє фінансування	1. Залучення додаткових інвесторів; 2. Краудфандинг
3	Технічний	Складність технологій	Інвестиції в технології
4	Кадровий	Відсутність кваліфікованих кадрів	Залучення фахівців зі сторони

Таблиця 4.7. Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Новий продукт	Вихід на ринок, Надання нових рішень	Розробка нового продукту; Вихід продукції на ринок; Захоплення максимальної долі ринку
2	Науково-технічний	Стрімкий розвиток технологій	Використання нових технологій
4	Монетизація реклами	Розміщення реклами на застосунку	Отримання доходу від розміщення реклами

Далі проведено аналіз пропозиції. Визначено загальні риси ринкової конкуренції (табл.4.8).

Таблиця 4.8. Ступеневий аналіз конкуренції на ринку

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	Тип конкуренції: монополістична	Існують конкуренти на ринку. Товар конкурентів принципово відрізняється від запропонованого товару	Розробка якісного продукту, щоб запропонувати альтернативні рішення клієнтам
2	Рівень конкурентної боротьби: світовий	Всі конкуруючі компанії знаходяться в будь-якому місці і пропонують свій товар в інтернеті для будь-якої аудиторії	Вихід на ринок з принципово новим підходом до рішення проблеми; Впровадження маркетингових стратегій на основних Інтернет ресурсах задля охоплення максимальної кількості потенційних користувачів; Роз'яснення серед користувачів унікальності продукту.
3	Галузева ознака: міжгалузева	Запропонований продукт може використовуватися в будь-якій сфері діяльності компаній	Розробка максимально зрозумілого інтерфейсу користувача; Врахування розмірів аудиторії; Відсутність концентрації на окремих типах клієнтів
4	Конкуренція за видами товарів: товарно-видова	Конкуренція між товарами одного виду.	Розробка функціоналу, що відрізняється від технологій конкурентів за своєю суттю.
5	Характер за конкурентними перевагами: не цінова	Не цінова – надання функціональності, що відрізняється від аналогів за своєю суттю.	Пропозиція можливостей, які не надають конкуренти.
6	За інтенсивністю: марочна	Наявність особливого знаку що відрізняє продукт від аналогічних продуктів	Впровадження унікальної назви. Реклама бренду

Далі проведено аналіз галузевої конкуренції за моделлю М. Портера (табл.4.9).

За результатами аналізу сформульовано висновок щодо можливості впровадження продукту на ринок з огляду на конкурентну ситуацію. Також сформульовано висновок щодо властивостей, які повинен мати проект, щоб витримати конкуренцію на ринку і зайняти конкурентну нішу.

Даний висновок враховано у процесі формування переліку факторів конкурентноспроможності. На основі дослідження конкурентного середовища, а також із урахуванням властивостей стартап-проекту (табл.4.2), вимог споживачів до товару (табл.4.5) та факторів маркетингового середовища (табл.4.6, табл.4.7) визначений перелік факторів конкурентноспроможності. Аналіз оформлюється за табл.4.9.

Таблиця 4.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Logaster	Існує вірогідність виходу нових конкурентів	-	Потребують унікальних логотипів з гарною якістю зображення	Частково присутні, можуть з'явитися нові
Висновки	Існує інтенсивна конкурентна боротьба в галузі	Слід враховувати можливість виходу конкурентів на ринок	-	Клієнти диктують умови функціонування на ринку.	Рішення існують, можуть з'явитися нові

Проаналізувавши властивості ринку з огляду на конкурентну ситуацію можна зробити висновок, що оскільки кожний з існуючих продуктів має свій підхід до вирішення задачі, то процес виходу на ринок представляється можливою, але слід враховувати, що рівень конкуренції високий.

Збір інформації про конкурентів проводиться з метою зробити власний проект більш конкурентноспроможним. При цьому є можливість з більшою упевненістю пропонувати свої послуги потенційним споживачам. Ця процедура необхідна тому, що можливі споживачі та партнери підприємства взагалі не знають, чим відрізняються діючі виробники на ринку.

Для успішного виходу на ринок продукт повинен робити акценти на своїй відмінності від конкурентів.

Таблиця 4.10. Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1	Web-application	Дає можливість використання програмного забезпечення будь-де для будь-якого користувача. Можливість кросплатформенного використання
2	Ціна рішення	Можливість генерувати логотипи безкоштовно
3	Унікальність товару	Унікальність логотипів необхідна для клієнтів
4	Різноманітність товару	Різноманітність логотипів є перевагою для клієнтів

На основі значених вище факторів конкурентоспроможності (табл.4.10) проведено аналіз сильних та слабких сторін проекту (табл.4.11).

Таблиця 4.11. Порівняльний аналіз сильних та слабких сторін

№	Фактори конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з запропонованим						
			-3	-2	-1	0	+1	+2	+3
1	Web-application	20	+						
2	Ціна рішення	17				+			
3	Унікальність товару	19				+			
4	Різноманітність товару	19			+				

Заключною частиною аналізу можливостей ринкового впровадження проекту є формулювання SWOT-аналізу (матриці аналізу слабких (Weak) та сильних (Strength) сторін, можливостей (Opportunities) та загроз (Troubles) (табл.4.11) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл.4.12).

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками

(прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

Таблиця 4.12. SWOT аналіз стартап-проекту

<p>Сильні сторони (S):</p> <ul style="list-style-type: none"> – Кроссплатформенність – Ціна рішення – Унікальність – Різноманітність 	<p>Слабкі сторони (W):</p> <ul style="list-style-type: none"> – Реклама – Сміслові навантаження логотипів
<p>Можливості (O):</p> <ul style="list-style-type: none"> – Відсутність схожих технологій у конкурентів 	<p>Загрози (T):</p> <ul style="list-style-type: none"> – Наявність конкурентів з альтернативними рішеннями

На основі SWOT-аналізу були сформульовані альтернативи ринкової поведінки, тобто перелік заходів, необхідних для виведення проекту на ринок та приблизний оптимальний час для його реалізації з огляду на можливість виведення конкурентами потенційних проектів.

Сформульовані альтернативи були проаналізовані з точки зору строків та ймовірності отримання ресурсів (табл. 4.13).

Таблиця 4.13. Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Розробка алгоритмів системи розпізнавання голосових команд та їх інтерпретація	20%	7-8 місяців
2	Відсутність реклами на перших етапах роботи рішення	80%	4-5 місяців
3	Розміщення рекламних постів у соціальних мережах, реклама по всім каналам	90%	2-3 тижні
4	Презентація товару на тематичних конференціях	80%	1-2 місяці

4.4.Розроблення ринкової стратегії проекту

Першим кроком розроблення ринкової стратегії є визначення стратегії охоплення ринку. Для цього проведено опис цільових груп потенційних споживачів (табл.4.14).

Таблиця 4.14. Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільового сегменту	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Малі компанії	Висока	Висока	Висока	Середня
2	Середні компанії	Середня	Середня	Висока	Середня
3	Великі компанії	Низька	Низька	Висока	Середня
4	Молоді компанії	Висока	Висока	Висока	Середня
5	Старі компанії	Низька	Низька	Висока	Середня

Згідно з проведеним вище аналізом можна зробити висновок, що цільова група, що підходить найбільше для поширення даного програмного продукту є молоді невеликі компанії. Такі результати отримано тому, що саме такі компанії схильні до використання новітніх технологій а також не мають коштів для закупки логотипів у спеціалізованих професійних компаній. Великі компанії, що давно на ринку схиляються до традиційних способів ведення справ. Далі у табл. 4.15 наведено визначення базової стратегії розвитку.

Таблиця 4.15. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Розміщення рекламних постів у соціальних мережах, реклама по всім каналам	Проведення реклами, освітлення унікальності рішення в інтернет мережі. Створення рекламних постів в соціальних мережах та в основних пошукових системах.	Унікальність технології запропонованої в рішенні. Відмінність від аналогічних рішень.	Стратегія диференціації

Наступним кроком є визначення стратегії конкурентної поведінки (табл.5.16).

Таблиця 4.16. Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, які?	Стратегія конкурентної поведінки
Ні, оскільки існують як конкурентні рішення, так і товари-замінники	Концентрація на існуючих клієнтів	Рішення не буде копіювати аналогів. Буде обрано власний підхід до рішення проблеми	Стратегія заняття конкурентної ніші

На основі того, які потреби мають споживачі з обраних сегментів щодо постачальника та його продукту (див. табл.4.5), а також в залежності від обраної базової стратегії розвитку (табл.4.15) та стратегії конкурентної поведінки (табл.4.16) розроблено стратегію позиціонування (табл.4.17), що полягає у визначенні ринкової позиції, за якою споживачі будуть відрізняти торгівельну марку/проект від інших.

Таблиця 4.17. Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Якість зображень	Спеціалізації	Використання передових алгоритмів	Якість, чіткість, осмисленість
2	Унікальність зображень	Спеціалізації	Відсутність подібних рішень	Відмінність, унікальність

Відповідно до приведеної вище інформації та проаналізувавши її можна зробити висновок, що як базова стратегія розвитку та як базова стратегія конкурентної поведінки для стартап-компанії найкраще підходить стратегія спеціалізації.

4.5.Розроблення маркетингової програми стартап-проекту

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у табл.4.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 4.18. Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Висока точність розпізнавання мовлення	Якість забезпечується передовими алгоритмами розпізнавання мовлення	Потрібно забезпечити якість взаємодії
2	Якість взаємодії користувача з інтерфейсом	Якість забезпечується розробкою якісних користувацьких інтерфейсів	Можливість керування не тільки голосом

Розроблено трирівневу маркетингову модель товару: уточняється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (табл.4.19).

Таблиця 4.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
1. Товар за задумом	WEB-додаток для генерації логотипів.		
2. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	Зручність та простота у користуванні	Нм	Ергономічні
	Унікальність зображень	Нм	Технологічні
	Кросплатформеність	Нм	Технологічні
	Якість: буде проведено тестування		
	Система; веб-сайт, що буде надавати доступ для системи.		
	Назва товару: NeuralLogo		
3. Товар із підкріпленням	До продажу: 1 місяць без реклами		
	Після продажу: постійне допрацювання алгоритмів		
За рахунок чого потенційний товар буде захищено від копіювання: унікальність технологій.			

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар, яке передбачає аналіз ціни на товари аналоги або товари субститути (табл. 4.20).

Таблиця 4.20. Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
0 грн	0 – 60000 / рік	10000000грн / рік	60000 / рік

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 4.21):

- проводити збут власними силами або залучати сторонніх посередників (власна або залучена система збуту);
- вибір та обґрунтування оптимальної глибини каналу збуту;
- вибір та обґрунтування виду посередників.

Таблиця 4.21. Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Вибір послуги на веб-сайті	Web-сайт	Виробник - клієнт	Web-сайт

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.22).

Таблиця 4.22. Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Клієнти обирають зручніший товар з потрібними функціями.	Веб-сайт	Унікальність	Демонстрація переваг для клієнтів.	Аргументація переваги над конкурентами, демонстрація результатів роботи.

Як результат створено ринкову (маркетингову) програму, що включає в себе визначення ключових переваг концепції потенційного товару, опис моделі товару, визначення меж встановлення ціни, формування системи збуту та концепцію маркетингових комунікацій.

Висновки до розділу

В четвертому розділі описано стратегії та підходи з розроблення стартап-проекту, визначено наявність попиту, динаміку та рентабельність роботи ринку, як висновок вказано що існує можливість ринкової комерціалізації проекту. Розглянувши потенційні групи клієнтів, бар'єри входження, стан конкуренції та конкурентоспроможність проекту встановлено що проект є перспективним. Розглянуто та вибрано альтернативу впровадження стартап-проекту та доведено доцільність подальшої імплементації проекту.

ВИСНОВКИ

Протягом виконання проекту проаналізовано вичерпну кількість матеріалів щодо історії розвитку технологій розумного дому, його роботів та інтернету речей, існуючих технологій та реалізацій розпізнавання й відтворення голосу, що дало нам змогу створити стійку, просту та легку для доопрацювання структуру програмної системи монолітного типу із підпроцесами, що реалізовуватиме інтерфейс взаємодії людини із машиною за допомогою голосу, що слухає, обробляє, інтерпретує та реагує у відповідь, і що працює у циклічному режимі, даючи змогу безперервно виконувати розпізнавання та інтерпретацію у подальші команди, а також спостерігати за станом робота, процесом розпізнавання за допомогою веб-інтерфейсу.

Описані вимоги до функціоналу системи розпізнавання голосових команд та їх інтерпретації. Алгоритм роботи системи робота-помічника описує повний цикл взаємодії з користувачем. Починаючи від того в якому режимі знаходиться робот в стані спокою, в нашому випадку це режим очікування кодового слова як сигнал про початок обробки голосу та порівняння отриманого з базою даних типових команд, закінчуючи тим що прописаний алгоритм поведінки в ситуаціях, коли система не змогла розібрати слово, або не знайшла відповідної команди.

Стійкість до помилок є важливим фактором при розробці подібних систем, адже існує багато сценаріїв в такій складній системі як «розумний дім», коли неправильна інтерпретація голосових команд може призвести до небажаних наслідків або до тупикових та невизначених ситуацій. Комунікація повинна бути також інтуїтивно зрозумілою для користувача.

Система повинна враховувати всі можливі помилки та правильно реагувати. Сповіщати користувача в момент переходів з різних станів, щоб уникати помилок. Також команди занесені до бази даних повинні значно відрізнятися між собою, щоб уникати небажаних спрацювань.

Засоби і бібліотеки, що ми застосували для розробки, дозволили суттєво спростити її, полегшити майбутню підтримку, додавання нових функцій, вбудовуваність у готові технічні комплекси. Також ми провели комплексну оцінку всіх параметрів техніки безпеки та охорони праці, отримавши всі необхідні параметри для облаштування комфортного та безпечного робочого місця для розробника системи. У майбутньому планується подальше розширення системи та доопрацювання наявних недоліків для досягнення максимальної ефективності та мінімальної вразливості до помилок.

Всі використані нами технології для реалізації модулів є сучасними та їх підтримка ведеться великими компаніями. Це дає впевненість у тому, що технології будуть продовжувати своє існування та будуть актуальними. Це забезпечить працездатність модулів нашого програмного забезпечення впродовж тривалого часу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Перспективи на ринку систем голосового управління [Електронний ресурс] // Хабрахарбр. – 2016. – Режим доступу до ресурсу: <https://habrahabr.ru/post/232613/>.
2. Семенюк О. В. Програмні засоби ідентифікації користувача за голосом на основі нейронної мережі / О. В. Семенюк, О. К. Колесницький // Науково-практична конференція «Сучасні тенденції розвитку системного програмування» (25-26 листопада 2016 р., Україна, м. Київ.
3. Picone J. Методы моделирования сигнала в распознавании речи / Joseph Picone, Ruslan Purov. - Кемерово. 2000.
4. Фланаган Дж. Анализ, синтез и восприятие речи / Дж. Фланаган. - М. : Связь, 1968.
5. Уоссермен Ф. Нейрокомпьютерная техника: теория и практика / Ф. Уоссермен. - Москва. – 1992. – 184с.
6. Восприятие информации в системах искусственного интеллекта: Учеб. Пособие / В.М. Игнатьев, Е.В. Ларкин. - Тул. гос. техн. ун-т: Тула, 1993.
7. Ахмед Н. Ортогональные преобразования при обработке цифровых сигналов / Ахмед Н., Рао К.Р. - М. : Связь, 1980.
8. Мясичев А. А. Управление голосом с помощью Android и Arduino [Електронний ресурс] / А. А. Мясичев. – 2015. – Режим доступу до ресурсу: http://khnu.km.ua/root/kaf/ksm/my_syte_g/.
9. Фаин В. С. Распознавание образов и машинное понимание естественного языка / В. С. Фаин. – Москва. : Издательство "Наука", 1987.
10. Любунь З. М. Основи теорії нейромереж: текст лекцій / З. М. Любунь. – Львів: Видавничий центр ЛНУ ім. Івана Франка, 2006. Алексеев А.В. Нейронные сети — М. : Издательство МЭИ, 2000. — 312.
11. Маркел Дж. Линейное предсказание речи / Маркел Дж. Д., Грэй А. Х. - М. : Радио и связь. 1980.

12. GritTec Speaker-ID: The mobile client - Программный комплекс голосовой идентификации [Электронный ресурс] – Режим доступа: http://www.grittec.ru/speaker-id_voice-identification.html.
13. Кузнецов В., Отт А. Автоматический синтез речи. – Таллинн: Валгус, 1989. – 135 с.
14. Пилипенко В.В. Алгоритм розпізнавання злитого мовлення з надвеликих словників із застосуванням вибірки інформації з баз даних/ "Міжнародний науково-учбовий центр інформаційних технологій та систем", Київ.
15. Vipul C. Rajyaguru Different Methods Used In Voice Recognition Techniques / Vipul C. Rajyaguru // International Research Journal of Engineering and Technology (IRJET) – 2016. - №7. – С. 699 – 702
16. Blind signal separation [Электронный ресурс]: веб-сторінка онлайн довідника Вікіпедія . – Режим доступу https://en.wikipedia.org/wiki/Blind_signal_separation.
17. Blind signal separation [Электронный ресурс]: веб-сторінка. – Режим доступу https://en.wikipedia.org/wiki/Blind_signal_separation.
18. Mel scale [Электронный ресурс]: веб-сторінка онлайн довідника Вікіпедія . – Режим доступу https://en.wikipedia.org/wiki/Mel_scale.
19. Filter Bank [Электронный ресурс]: веб-сторінка онлайн довідника Вікіпедія . – Режим доступу https://en.wikipedia.org/wiki/Filter_bank.
20. 5 Benefits of a 3-Tier Architecture [Электронный ресурс]: веб-сторінка. / Izenda, Inc // – 2016 – Режим доступу <https://www.izenda.com/blog/5-benefits-3-tier-architecture/>.
21. RESTful API [Электронный ресурс]: веб-сторінка. / Mary Rouse // – 2016 – Режим доступу <https://searchmicroservices.techtarget.com/definition/RESTful-API>.

ДОДАТКИ

ДОДАТОК А
ІЛЮСТРАЦІЙНІ МАТЕРІАЛИ

Підсистема аналізу та інтерпретації голосових команд

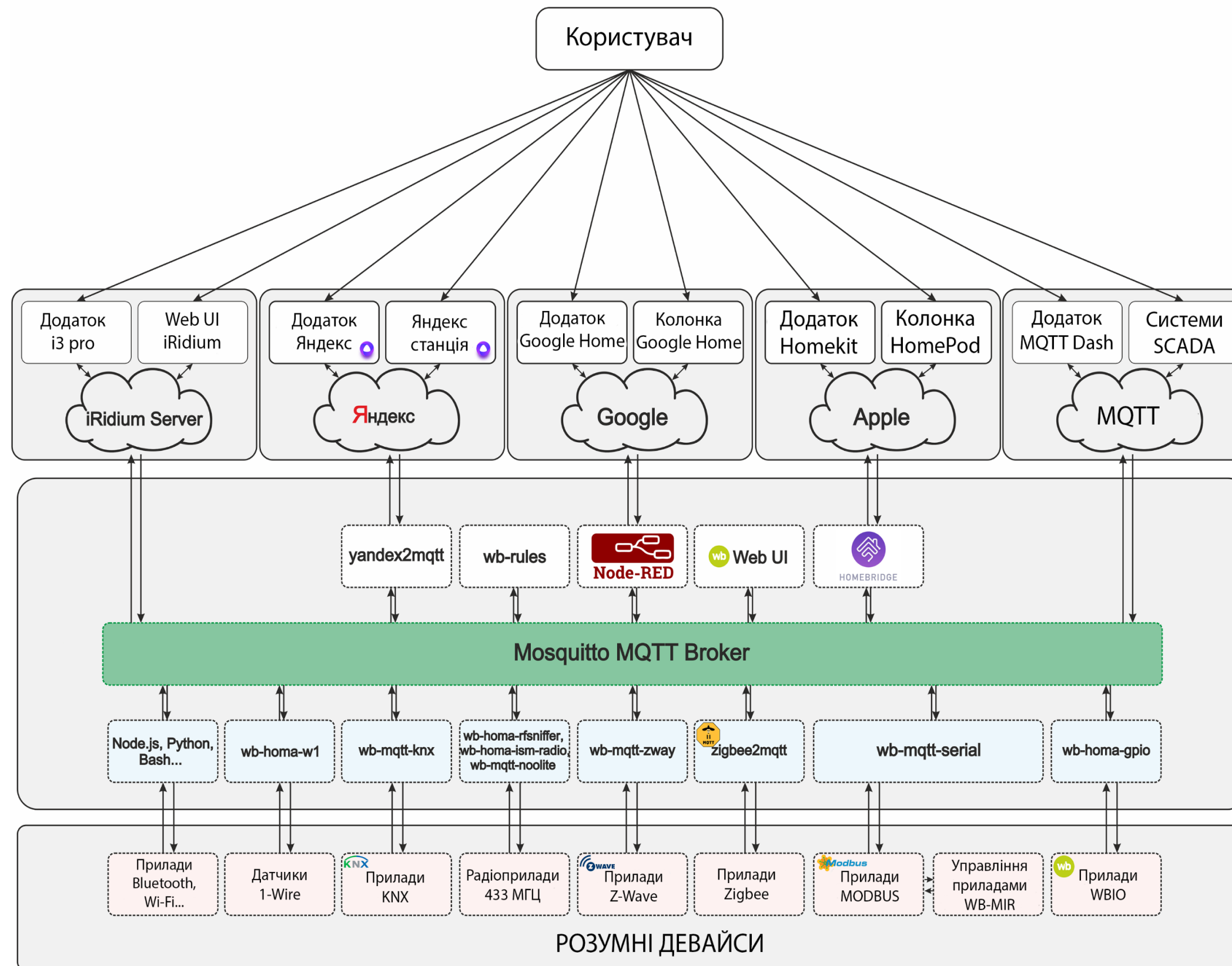
Система розумний дім



Демонстраційний плакат №1
до магістерської дисертації на тему
«Система інтелектуального аналізу голосових команд, їх інтерпретація
для подальшого планування дій робота»

Розробив: студент гр. ІК-82мп Менчинський О.С.
Прийняв: к.т.н., доцент Тимошин Ю. А.

Приклади реалізації взаємодії з голосовим управлінням

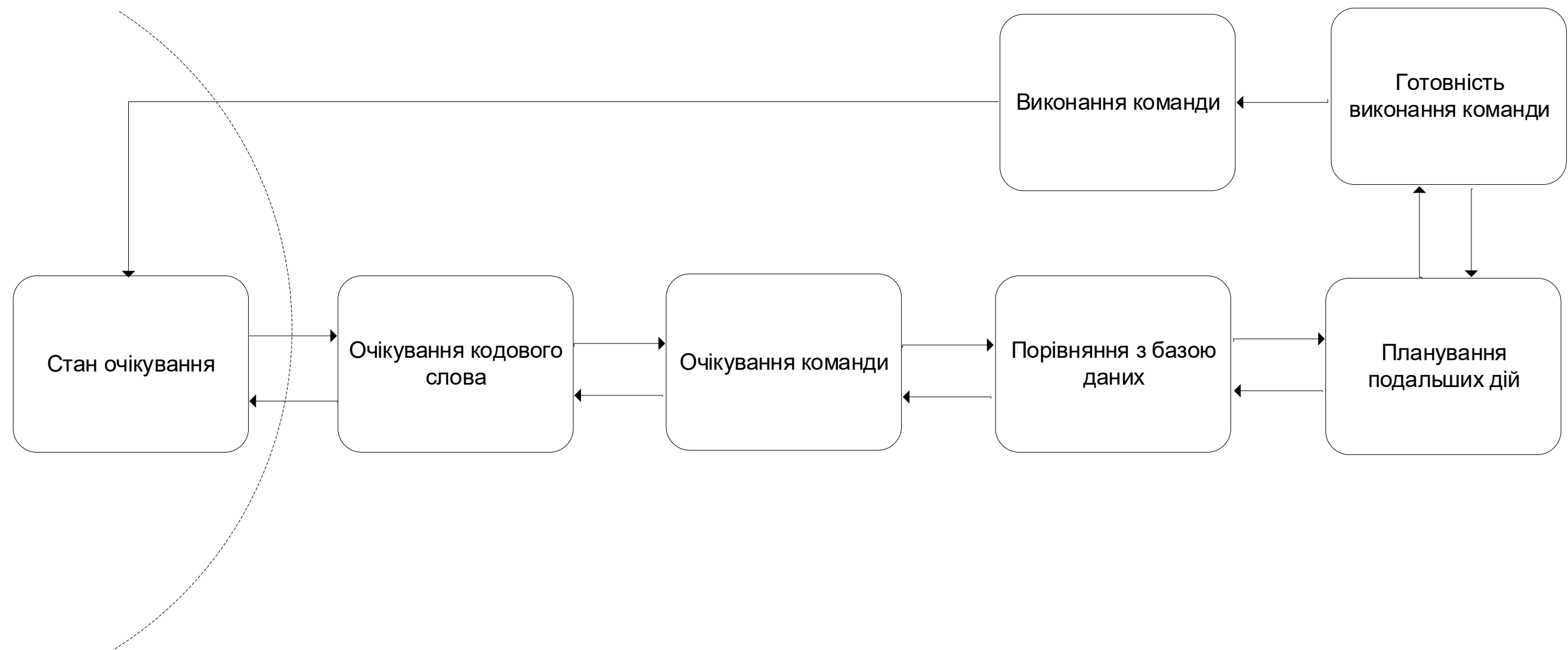


Демонстраційний плакат №2
до магістерської дисертації на тему
«Система інтелектуального аналізу голосових команд, їх інтерпретація
для подальшого планування дій робота»

Розробив: студент гр. ІК-82мп Менчинський О.С.

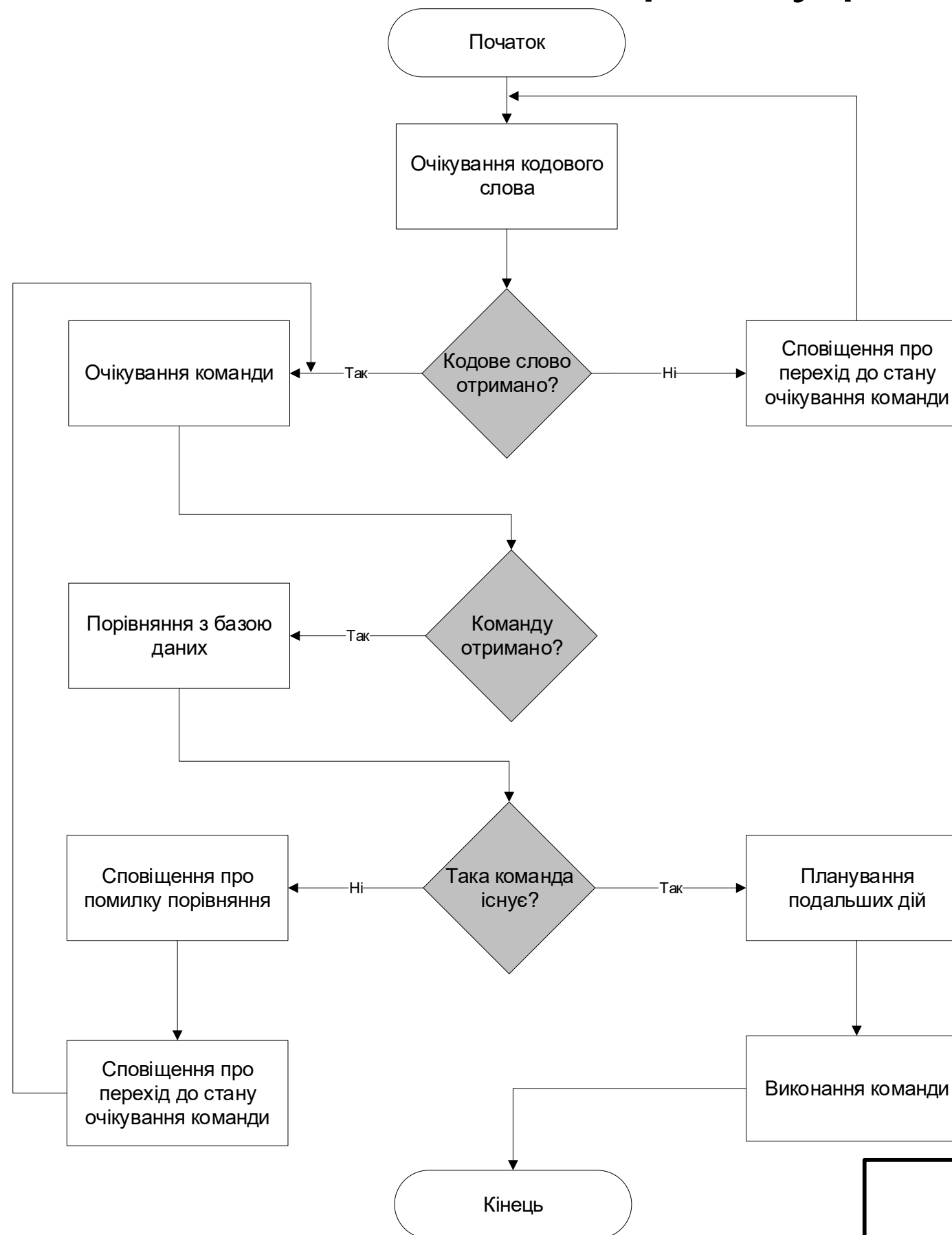
Прийняв: к.т.н., доцент Тимошин Ю. А.

Стани системи аналізу та інтерпретації голосових команд



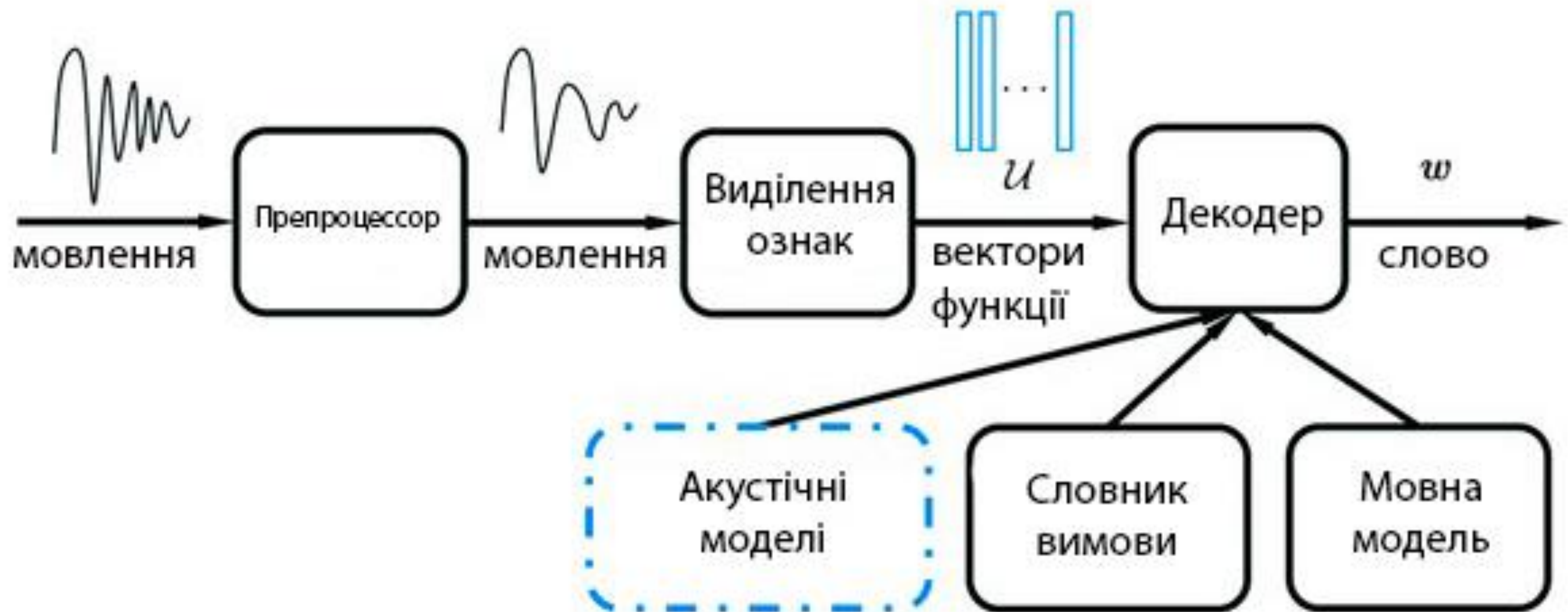
Демонстраційний плакат №3
до магістерської дисертації на тему
«Система інтелектуального аналізу голосових команд, їх інтерпретація
для подальшого планування дій робота»
Розробив: студент гр. ІК-82мп Менчинський О.С.
Прийняв: к.т.н., доцент Тимошин Ю. А.

Блок-схема алгоритму роботи системи



Демонстраційний плакат №4
до магістерської дисертації на тему
«Система інтелектуального аналізу голосових команд, їх інтерпретація
для подальшого планування дій робота»
Розробив: студент гр. ІК-82мп Менчинський О.С.
Прийняв: к.т.н., доцент Тимошин Ю. А.

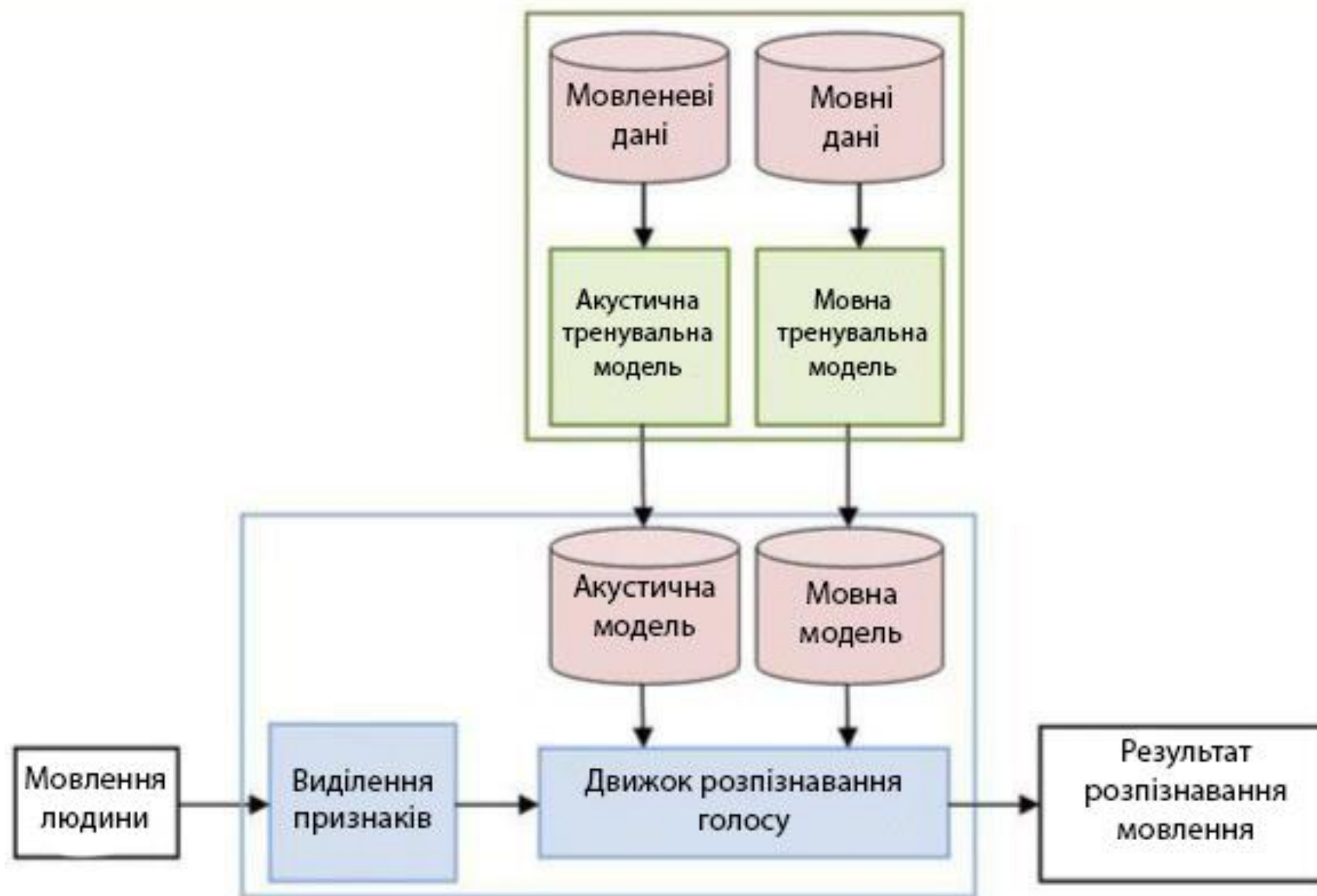
Схема автоматичного розпізнавання мовлення



Демонстраційний плакат №5
до магістерської дисертації на тему
«Система інтелектуального аналізу голосових команд, їх інтерпретація
для подальшого планування дій робота»

Розробив: студент гр. ІК-82мп Менчинський О.С.
Прийняв: к.т.н., доцент Тимошин Ю. А.

Архітектура системи розпізнавання мовлення



Демонстраційний плакат №6
до магістерської дисертації на тему
«Система інтелектуального аналізу голосових команд, їх інтерпретація
для подальшого планування дій робота»

Розробив: студент гр. ІК-82мп Менчинський О.С.
Прийняв: к.т.н., доцент Тимошин Ю. А.

ДОДАТОК Б
ПЕРЕВІРКА НА СПІВПАДІННЯ